Development of API for an IoT management platform

HAMK
HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

Bachelor's thesis

Valkeakoski, Automation Engineering

Autumn 2018

Illya Nizyev

HÄMEEN AMMATTIKORKEAKOULU
HÄME UNIVERSITY OF APPLIED SCIENCES

ABSTRACT

Automation Engineering
Valkeakoski

Author Illya Nizyev **Year** 2018

Subject Development of API for an IoT management platform

Supervisor(s) Juha Sarkula

Commissioned by APInf Oy

ABSTRACT

The objectives of this thesis included examining the current state of the Internet of Things world, outlining existing implementations, pointing out current and potential future problems there and designing own example implementation by developing the API for IoT management platform (Apinf).

For the part of creating the API objective was achieved by first creating the required software part of the monitoring feature and then the API to acquire data. An API stands for application programming interface. This interface makes it possible to exchange data and content between different applications or hardware. The API of the application is fully based on the data it can produce. A modern development project of software or hardware has to include an API, so that there is always a more efficient method for to data to control parts of the application. The theoretical part of the objectives was achived by acquiring academic literature on IoT and by analysing related research works.

Keywords Internet of Things, IoT, API.

Pages 30 pages including appendices 9 pages

CONTENTS

1 INTRODUCTION

The Internet of Things (IoT) is a term, which has already become a very important topic not just in professional circles, but in the everyday life of ordinary consumers. This technology is taking by storm people's houses, industrial parks, factories, infrastructures of cities and a number of other important spheres of the modern world.

The Internet of Things embraces technical, social, and related fields, making them co-exist and evolve together. Industrial devices and tools, consumer goods, cars, everyday things are now connected to the Internet. It opens tons of possibilities for powerful data analytics collection, remote controlling and changes the way we work, live and operate everyday devices.

At the same time, the Internet of Things can leave open questions in security, efficiency and legitimacy areas, which raises considerable challenges that could slow down the development of this topic.

A high number of IoT technologies promise to change the way people live. As for consumers, IoT offers a variety of products such as home automation, energy control and other Internet-based devices. In addition, it enables different personal devices like medical monitoring, fitness tracking for everyday usage. IoT transforms people's everyday routine in analytics, which helps companies to improve their products remotely and opens new never analysed before subjects of people's life.

IoT transforms the way cities are operated. It enables systems which make city "smart", this includes traffic controlling, water, energy, surveillance, emergency alarms, pollutions and many other.

A high number of companies, and organizations are embracing this technology in theirs processes. This allows using available resources outside of accepted frames, connecting to the Internet every part of the companies' production stage. Generally, the IoT is about how people and technologies interact with the Internet in terms of personal and working areas.

This thesis introduces the world of IoT, describes the current state of Internet of Things and in the practical part of this thesis explores the technical aspects of IoT through APIs. API is the most efficient and fundamental method of communication with a software or hardware system. In other words, API is the modern language of a computer system.

The theoretical part of the thesis is divided into three parts. The first part provides a theoretical background on the topic of the Internet of Things. It includes a definition of the term IoT and briefly describes its history. The second part provides examples of IoT implementations, describes its purposes and the technical aspects of APIs. The third part addresses current and future issues, and mostly focuses on problems which may be occur with the expansion of the IoT.

The empirical part discusses a working API of the IoT service known as Apinf API Management System. This system consists of JavaScript, HTML and CSS code that gives access to one of the functional blocks of the software system of Apinf. The current report describes the method of retrieving data about the availability of APIs.

2 BACKGROUND

2.1 Terms definition

One of the first discussions on the idea of a network consisting of smart devices started back in 1982, when the modified ice-cooling machine was the first device connected to the Internet in the Carnegie Mellon University. The machine was able to give information on the number of drinks loaded into it, and also when the loaded beverages were cooled enough. (Becomes the Stings, n.d.)

The term "Internet of Things" was first used by Peter T. Lewis in his speech of 1985 during the session of the Federal (FCC), in support of wireless communication at the 15th Legislative Conference. He stated that “the Internet of Things is the integration of people, processes and technology, with connected devices and sensors to provide remote monitoring, status, manipulation and evaluation of trends in these devices”. (Becomes the Stings, n.d.)

The concept of the Internet of Things became popular in 1999, thanks to the Centre of Auto-identification at the Massachusetts Institute of Technology and publications in the field of market analysis. Kevin Ashton (one of the founders of the centre of auto-identification) saw the radio frequency identification as a background for the beginning of the Internet of things, although he preferred the term “the Internet for things”. (Becomes the Stings, n.d.)

IoT is interworking of physical devices, vehicles (also called "Connected devices" or "smart devices"), buildings and other items equipped with electronics, software, sensors, and network connection that allow these objects to collect and share data.

In 2013, the Global Standards Initiative on Internet of Things, IoT-GSI defined the Internet of things as "infrastructure of information society". Internet of things allows devices to be detected or controlled remotely through the existing network infrastructure. It creates opportunities for direct integration of the physical world into the computer system, resulting in an efficiency, accuracy, economic benefit and to limiting of human intervention for preventing unpredictable errors. (A Brief History, n.d.)

When the Internet of things is supported by sensors and executive mechanisms, this technology becomes the foundation for more common cyber-physical systems, which creates such technologies as smart networks, smart houses, smart transport and smart cities. (An IoT system, n.d.)

Each thing of the IoT is uniquely identified through the built-in computing system but is also can be controlled and interact through existing infrastructure of the Internet. According to experts, by 2020, the Internet of things will include almost 50 billion devices (An IoT system, n.d.)

Industry 4.0 is relatively new term, which usually refers to a new phase of the industrial progress. Industry 4.0 can be described in a similar way as the Internet of Things, but more in a reference of industrial terms and ideas. The fourth industrial revolution, or Industry 4.0 is a continuation of the industrial evolution. This revolution brings the Internet, big data and everything that produced by these technologies to the existing industry. (What is Industry 4.0, n.d.)

3 EXAMPLES OF EXISTING IMPLEMENATIONS

3.1 Smart Home

A smart home is a combination of different electronic devices that could be distantly controlled by a remote control through a common network connection. For this to be possible, all devices have to have internet connection. Smart home is often called a smart house, adaptive house, home automation, and intelligent house. The purpose of smart home system is to improve quality of life by implementing electronic devices that manage appliances and other assistive services. Smart home devices can be used for limitless purposes, some include control of home security, healthcare, control of indoor climate, control of lights, safety, and energy conservation. One of the most commonly used smart home appliance is the remote surveillance systems that use web and telecommunication technologies to remotely control and monitor home. (Smart home, n.d.)

Smart home devices and software are often based and strongly connected to standards of the producing company or brand. These standards could include simple instructions or usage examples, but more often device or software behaviour is strictly defined and scripted. When it comes to the smart home APIs, companies tend to write best practices instructions rather than open and free documentation. These practices are due to safety and privacy reasons. Lack of freedom in smart home API development usually is complemented by the variety of software and hardware on a free market. Users are free to buy and combine different pieces of software and hardware into a smart home solution. This approach can give ability not only produce good API documentation but also include more features compare to big companies' brands. A good example of such system can be one produced by the USA company by the name Nest. The aim of their products is to create a connected home, but they are very limited in API functionality. (Smart home, n.d.)

Usually smart home devices have an ability to self-educate and improve, meaning that they are able to analyse repeating patterns of owner's behaviour and as a result making corresponding adjustments to its system. A real-life example would be a smart home device that controls lights. At first, it studies owner's timetable and use of lights in the house and then makes improvements that allow the reduction of use of electricity and therefore result in monetary savings in long term. Other practical implementations may include registering any motion that should not be there while the owner is away and automatically informing security services or fire department in case of emergency. (Smart home, n.d.)

In terms of types of smart home appliances, they can be classified as hardwired or wireless. Both of them have their advantages and disadvantages. Typically, a wireless set up is less expensive and simpler to install, it can be easily dismounted and re-installed at a different location. On the down side, they may be considered less secure and less reliable than their counterpart. The hardwired set ups tend to be costlier, better protected against external threats, rarely crash and cannot be removed. This can be seen both as a benefit and as a disadvantage because on one side it means it cannot be dismounted and re-installed when moving to a different location, however, it increases the monetary value of a building at a sale. For a price reference, wireless smart home system with smart lights, smart temperature control and smart security and surveillance may cost several thousand euros. On the other hand, an identical in terms of features hardwired system can cost three or four times more the amount than the wireless option. (Smart home, n.d.)

3.2 **Wearables**

Smart wearables are designed to be worn on the body for the purpose of keeping constant contact with it so that device could analyse the person

and/or the surrounding. There are mainly two ways of categorising connection communication types of wearable, which are either independent wireless connection or through a separate device (i.e.: a mobile phone). After the wearable device reads and collects information about its owner or the surrounding, it analyses the data either on its own or transfers the data to an external device that processes it. As an outcome, the user can see evaluated results. Almost all smart wearables have functions of storage, communication, activation and control. (Wearable Technology, n.d.)

In the modern world, smart wearables provide unlimited and very useful applications to the daily life of the world. Areas of personal healthcare, emergency prevention, work safety, productivity improvement, emergency treatment at private and public locations. Some consider the ability to constantly record owner's movement, changes in the surrounding area and biological signals to be the most important functions of wearable technology. The most popularly used smart wearables in the modern world are smartwatches and fitness trackers. Nevertheless, there is huge potential in devices that are still in the process of early development. Some examples are intelligent patches, textiles, glasses, VR headsets, exoskeletons (wearable robotics), and intelligent jewellery (i.e. earrings that assist in hearing and other small devices that analyse sleep levels). (Wearable Technology, n.d.)

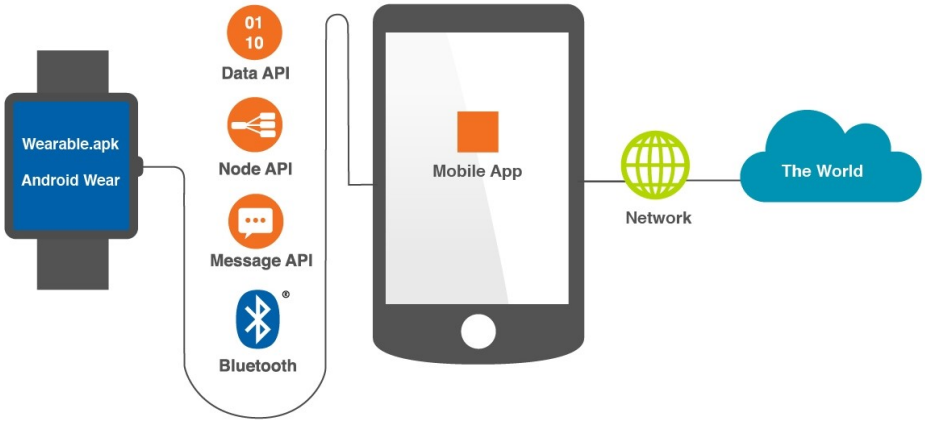

Figure 1. Example of wearable architecture (Wearable Technology, n.d.)

Moreover, most developed governments admit that smart wearable technology will be revolutionary and has potential to change the way humans are going to live. Combination of wearables with other intelligent devices and the newest developments in the Internet of Things (IoT) have potential to innovate interaction between people and the digital universe. This seems most possible with smartphones. As of today, the intelligent smart wearables market is enormous (some estimate yearly revenue to be

around thirty billion dollars) and it continues to grow. (Wearable Technology, n.d.)

Leaders in the field of wearable technology, such as Apple watch, Samsung Gear, Garmin compete to provide customers the best products with the most useful applications. Figure 1 shows the connection of a wearable device with a mobile app on the phone. This connection describes current state of wearable devices - they are produced mostly by big companies with strong smartphone dependencies. The main purposes are to produce very mobile wearable devices that are small, enduring and highly efficient. (Wearable Technology, n.d.)

3.3 **Connected cars**

The automotive industry has potential to become the most prominent part of the Internet of Things, a concept where every gadget is connected to the Internet. When more and more devices do that, the number of applications and software for the vehicles increases. Every car is going to have motion sensors all throughout the vehicle's body. The automobile industry is definitely exploring and implementing self-driving car features. The most recent development in this direction is often called Vehicle-to-Vehicle or Vehicle-to-Everything. The point of this idea is to set up and systemise connection between the automobile and other devices or so-called "things". Examples of such "things" include other cars, cloud storage, mobile phones, data storage, roads, etc. (Connected Car Architecture, n.d.)

Smart self-driving cars have various benefits as they are initiative, coordinated, intelligent and are primarily designed to improve overall safety on the roads with features like collision detectors, automatic lane switcher, and coordinated movement with other smart vehicles. Moreover, smart vehicles are going for intelligent transportation with features like route planning according to the traffic schedules and constant connection to the Internet would be a must-have. This would make traveling a lot faster and more efficient and would eventually result in saving money in long-term. (Future connected vehicles, n.d.)

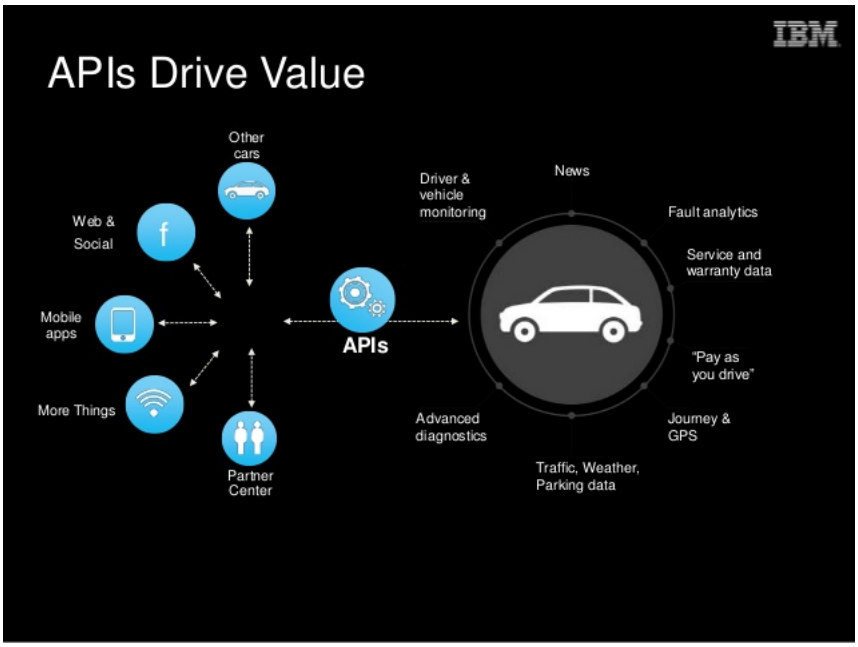

Figure 2. Connected cars scheme (Connected Car Architecture, n.d.)

The smart automobile industry is growing at a very high pace and according to some estimates, it is going to reach more than 100 billion dollars in revenue in 2019. The more finances flow into the industry, the more progress and development will be created. Furthermore, there is a lot of room for optimizing road safety and its efficiency. The problem of traffic is global, and it is only worsening day by day, which negatively affects the economy and ambient environment. Smart vehicles have potential to provide solutions to unending traffic jams through implementing smart traffic control systems in cars. However, for all of this to be possible, every single smart vehicle has to have a good quality internet connection on the road. This is actually possible in more economically developed countries and it lets drivers use various security-related applications that prevent cars from theft, could track the car movement if stolen. (Connected Vehicles, n.d.)

The issue of sharing information between cars is not very complicated from a technical point of view for the manufacturers, but it is complicated on the level of consumer privacy and security. Building connection between hardware parts of car and software applications usually blocked by absence of proper integration from third-party components. The API standards, Figure 2 shows how crucial and important APIs are, developed by more modern companies should help to improve integrations. To tackle this potential problem, companies have to ensure that security, protocol encryption, authentication and protection of data is at the highest level possible. (Future connected vehicles, n.d.)

3.4 **Smart cities**

A district that utilises information and communication technology to improve its operational efficiency, to present information to people and to improve the overall well-being of citizens and governmental ministries is called a smart city.

Another definition of a smart town would be a city that has developed infrastructure that provides a basic quality of living, an eco-friendly ambient environment by using various smart technologies. Other aspects of city life that could be improved include electricity distribution, sewage management, public transportation, a ubiquitous connection of all things through the internet, security, surveillance, safety on the streets, etc. (What is a 'smart city', n.d.)

Due to its nature, it is quite challenging to come up with a definition that would include every possible aspect of a smart city. It is easy to assume that smart city is related only to intelligent technology, which is true in a way, however, it is not limited to it. Technology is the main instrument that improves the management of cities by creating smart transportation, energy, infrastructure, etc. As mentioned earlier, a smart city is not limited to digital technology. It consists of different aspects, too. Smart cities aim to connect people and business capital through various implementations of ICT. (Smart cities, n.d.) As a result, this should create an environment where society, business, and government are all linked together and are working to achieve a shared objective to make the city more productive, wealthy, effective, and sustainable. (Smart city, n.d.)

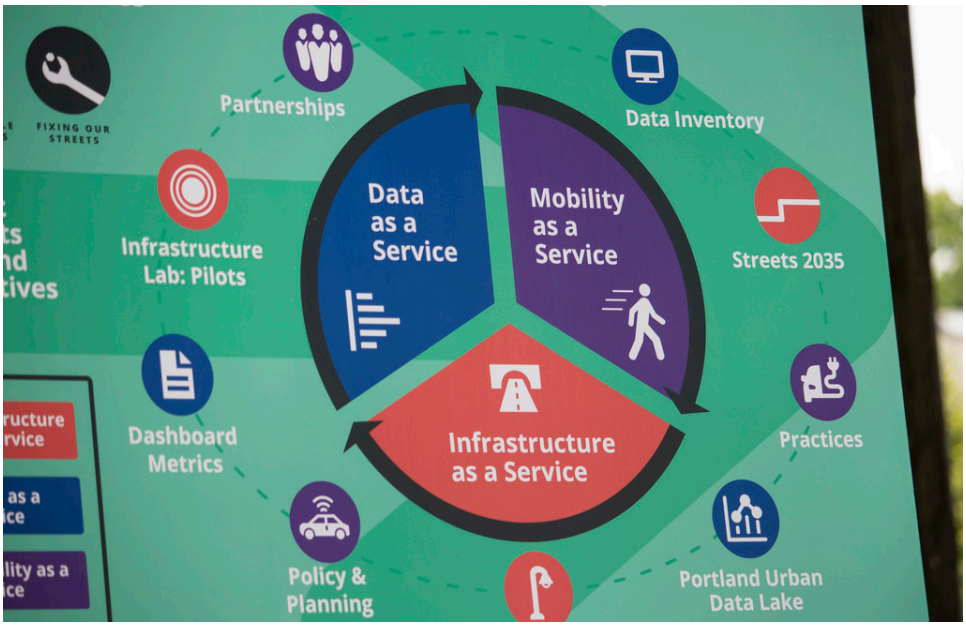

Figure 3. Smart city distribution chart (Smart city, n.d.)

To sum up all of the above, smart city can stand for various things, however, the main purpose remains the same. It consists of using

information and communication technology through the internet to tackle daily urban problems. (Challenges for smart cities, n.d.)

The importance of implementing such technologies and transforming our cities into smart cities is a logical response to a global urban growth. More and more people move into the cities and therefore urban population grows at a very high pace. Along with a lot of opportunities, there are various problems in the megalopolises, including unemployment, distribution of resources, environmental issues, traffic congestion, and overall social resilience. (Smart city, n.d.)

APIs play an important role of smart city distribution data (Figure 3). Every city consists of many different resources of data, which should not only be properly distributed but also protected. That is when APIs come to play. APIs help city data infrastructure be well accessed and understandable. City with good written API gives developers and users access to its data to improve lives of citizens. (Driving Smart Cities, n.d.)

Smart cities should be designed for the overall benefit of the society. There is a danger when smart cities are falsely given objectives of solely economic growth. For the benefit of the society, the smart city movement should not forget the issues of rights, justice, and citizenship.

3.5 **Agriculture**

The agriculture industry is in the process of transformation and will become more important than ever before in the nearest future. The latest UN Food and Agriculture Organization report says that people will need to increase production of food by 70% in 2050 in order to fulfil the rapidly growing population of the planet. Farmers and agriculture industry are starting to find solution to meet this demand by using the Internet of Things to improve and analyse their production line and capabilities. (Big data & smart farming, n.d.)

Last several decades, around the world, there has been a dramatic increase of awareness about the food safety and there has been a rise for improvement of food quality. These reasons made many countries and people in our society to create and adopt new ways to change their farms and fields into more automated and efficient ones. (Big data & smart farming, n.d.)

The idea of improving farms, fields and the agriculture industry is not new. Different approaches were already existing and implemented hundreds of years ago, for instance, introduction of the cotton gin instead of old handheld tools in times of the Industrial Revolution. (Big data & smart farming, n.d.) During 1800s many new tools like the first gas-powered tractor, grain elevators and chemical fertilizers were invented and popularized among farmers and factories. In late 1900s farmers were

introduced to methods of using satellites to help them in planning of their field work.

Good examples of the IoT applications in agriculture industry are livestock data gathering, farm vehicle monitoring, storage analyse, animal tracking and many more. Different kind of sensors may be installed in the ground or placed on animals, data is sent to configured servers, then farmers connect to them via the Internet and access all needed information using mobile phone or computer (Figure 4). This way the can always know what is happening in every place of their farm, that is especially helpful for big farms. (Big data & smart farming, n.d.)

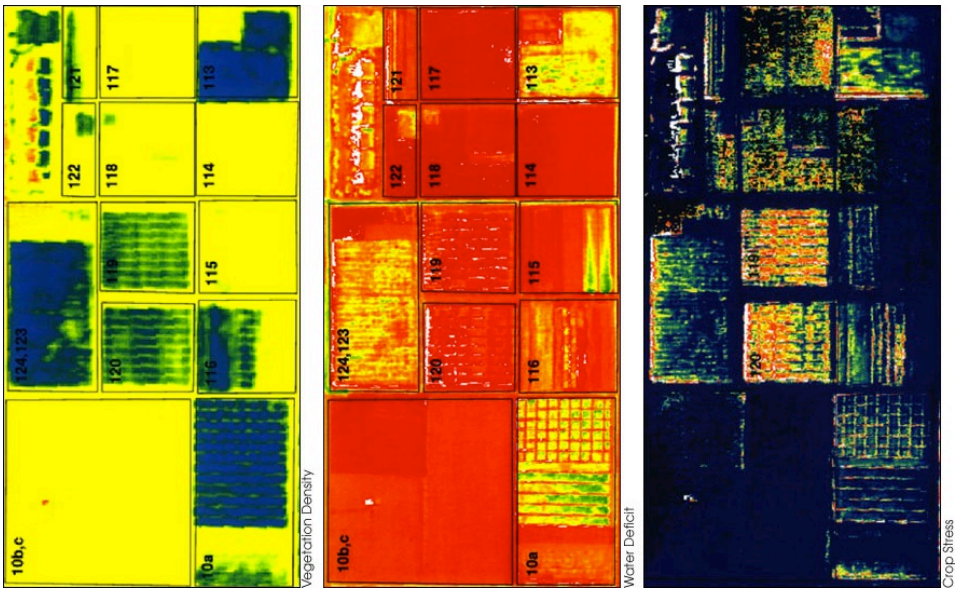

Figure 4. Remote sensing in precision farming (Big data & smart farming, n.d.)

The IoT is changing almost everything in modern farming and agriculture, bring the future of the industry to the next level of efficiency. Farms are already adopting and to the fast changes that brings the Internet using new technologies as drones and sensors. Some of these new technologies are becoming standards of the industry making every farm a smart farm. Farmworkers have already started to use different high-tech techniques and methods to improve and increase efficiency of their everyday work. (Big data & smart farming, n.d.) For instance, sensors installed in fields help farmers to build maps with both resources and topography of the area, including such elements as acidity of soil and its temperature. Also, they now are able to predict climate forecasts and create daily, weekly or monthly weather patterns.

As it was already said, farmers now can use their smartphones to remotely check their animals, equipment and livestock to get important information, such as time for feeding or issues in the fields almost immediately. ((IoT) for Agriculture, n.d.) They don't need any more to be

in the fields all the time, some actions can be delegated to automated systems. Also, farmers can now use modern technologies to run predictions of their livestock and crops to even more automate process of feeding. Another great example of smart agriculture application are farm drones, which can help to distribute time for efficiently and have less human resources to oversee bigger fields. ((IoT) for Agriculture, n.d.)

Although benefits are clearer, there exist some challenges and biggest of them is the cost of equipment, which in addition usually requires the wider internet connection to cover all territory.

3.6 **Smart Retail**

The Internet of Things has different applications in the area of retail business. Some of the existing technologies consists of RFID tracking devices, infrared visitor-traffic counter, wireless and cellular tracking applications, digital banner, etc. These smart devices can be used in supply chain, staying in touch with the customer, and other smart retail applications. Here is a more in-depth description of reasons and purposes for using smart retail. (IoT applications retailers, n.d.)

One of the most commonly used technologies for storage and inventory management lets owners control the energy distribution and use, predict equipment crashes, or warning about other problems. Nowadays, there are complicated inventory and machinery in almost every shop because refrigerators are a must-have. (the IoT applications retailers, n.d.) To maximise efficiency of fridges, they could be equipped with temperature monitoring and controlling devices that would protect food from being spoiled and at the same time maintain low power consumption rates.

Transporting inventory efficiently is always a priority of smart retail, and the Internet of Things is the main factor that makes it happen by improving the general vehicle maintenance, tracking and route planning. The Internet of Things can help retail owners know precisely the distance from pallet of merchandise to a certain shop. (IoT applications retailers, n.d.)

Smart retail can also offer useful improvements to warehouse management by applying inventory automation, and robotics that is controlled remotely through the internet. The Internet of Things can help track and analyse sales figures live. The use of such technology will eventually change modern definition of a warehouse, because it will completely automated and self-managed according with the demand. (IoT applications retailers, n.d.)

It is a common practice today for customers to check the prices from their smartphones. This can be used for retailers' advantage by using principles and methods of the Internet of Things. (Everything for Cities, n.d.) For example, by creating a customised offer list or provide location-specific

offers straight from the shop. This is achieved through analysing customer's identity, time and place of the request and responding accordingly.

Another useful feature of a smart store is analysing visitors' habits, hours of visiting, number of customers, rush hours, etc. This can be achieved through the use of surveillance cameras and the footage can further studied for the benefit of the business. For example, shelf-layout could be adjusted to maximise sales. The number of ways the collected data could be used is unlimited in terms of marketing. (IoT applications retailers, n.d.)

The differentiation with the Internet of Things will come from a retailer's ability to sense, understand and act on the Internet of Things data with analytics. To take act reasonably in this potential area, retailers should focus on smart store applications that improve customer's experience and create value.

3.7 **Healthcare**

Another example of implementation of the Internet of Things in modern world can be found in the healthcare, from smart sensors, medical applications to remote monitoring. It aims to improve work of medical workers to deliver more accurate results, and also to keep patients healthy and well informed. The IoT with healthcare, patients are more engaged in the process of their treatment and can save time by having less interactions with their doctors. (Health Monitoring and Management, n.d.)

Modern hospitals can be described through their applications of the IoT in day-to-day activities. Some hospitals are nowadays focusing on improving communication with their patients providing more detailed and relevant information, where other taking advantage of the IoT applications in keeping track of inventory. The IoT has a potential to impact every aspect of healthcare, automating more and more processes. Applications of the IoT help to optimize and improve surgical operations, inform family and relatives about the results and steps of the operations, alert about critical situations or successful results. Specific implementations of the IoT applications provide protection to new-borns and allow doctors and nurses to get updates and alerts on any NICU patients' changes and especially vital changes. Using wearables has encourages doctors to use the IoT to engage with patients, monitor remote patient and be ready to help with advice without leaving hospital. The IoT can be used to collect and analyse data from remote and in-hospital patients to predict the state of health and improve methods of treatments. (Health Monitoring and Management, n.d.)

Recent more advanced technologies and more agreeable patient attitudes bring more potential into the future of the health related IoT implementations. Hardware and software companies spend more

resources and efforts on improving usability, quality of products, including new processes of making sensors, batteries, devices smaller but more efficient. Personal data sharing is becoming more secure and easily accessible by patients and doctor. These changes of health industry and other contributions to the field helps with the expansion of the IoT. The IoT in healthcare provides many advantages to society. People become more aware of their health and tend to be more engaged with doctors and own healthcare. As a result, we get more informed and self-aware society. (Health Monitoring and Management, n.d.)

These health related IoT solutions are wide-ranging and include treatment testing, medical devices upgrades, and especially relevant in modern society fitness monitoring, which is a bridge between commercial and medical wearables. Together with various applications and devices of IoT, medical and social connections are improving and increasing. (A guide to healthcare IoT, n.d.)

The Internet of Things allows to record, keep and use health related information continuously and it also provides opportunities for this data to be personalized and applied immediately. (A guide to healthcare IoT, n.d.) It is becoming more and more evident that our modern type of healthcare is going to be changed radically from hospital-centred to eventually home-centred in the coming decades.

Despite unlimited and positive ways, the IoT could be used in medicine, it does have its obstacles and challenges. Before implementing any software and technology that would collect massive amounts of data about users, hospitals, IT companies and governments have to ensure maximum levels of data security and protection. This is just one example of a challenge that is in the way of implementing the IoT into the healthcare system. However, this barrier might only slow down the unavoidable revolution that the IoT is going to create in the world of medicine. (A guide to healthcare IoT, n.d.)

The Internet of Things healthcare gadgets are very common in hospitals, but they may struggle to stay up-to-date and patched, especially after they get out of the hospital network. Fragility of data sharing and the dangers of cyber-attacks is, nevertheless, a real issue for medical providers, industry and the society. Consumer data and trust should not be compromised. (Health Monitoring and Management, n.d.)

3.8 **Industrial Internet**

The Industrial Internet of Things (IIoT) mostly referred as the Industrial Internet, puts together analytics, machines and working people. It's the network of connected communicating devices, which make possible to monitor, analyse, gather, exchange and deliver important and valuable information in more efficient manner then before. This information helps make more clever and faster decisions in industrial business environment.

(industrial internet, n.d.) The IIoT is changes the industries like never before. It enables predictive analytics to find corrosion inside of pipes or gives real-time data to indicate additional capacity in a plant or to help preventing cyber-attacks and so much more. The IIoT with its' software solutions are making possible archiving high and successful business results. (industrial internet, n.d.)

The IIoT opens opportunities for industrial companies to increase efficiency in such fields as power, energy, gas and oil, healthcare, manufacturing and many other. This lead to transformation of industries and significant financial profits. The IIoT drives extraordinary levels of competence, capacity and performance. For the most parts it can be achieved by combining machine-to-machine (M2M) exchange, big data analyses, cyber security, HMI etc. (IIoT: An analysis framework, n.d.)

The idea of "Industrial Internet" has as of late risen as a subject of high attention in the business both globally and in particularly in Finland. While exploring ideas and technologies of already well-known subjects, for example, "Internet of Things" and "Cyber-Physical Systems", the level of interest of the idea that the Internet can improve industries is nowadays has been significantly raised. The assembling enterprises base increasing efficiency with more advanced technologies, for example, sensors, actuators, remote systems, cloud technologies, computational modelling and recreation, and portable UIs and many more. To sum up, industries are highly interested in bringing manufacturing ventures and their ecosystems to modern Internet age. (IIoT: An analysis framework, n.d.)

The IIoT incorporates the use of data and communications technologies across the industries within the society to enhance the proficiency of activities and the production to create new values for clients and users. Internet and portable technologies, such as mobile for instance, have changed to be more effectively accessible devices, which are less complex and less expensive and help to create and develop new applications. Through their utilization, the Industrial Internet empowers new industrial and administration organizations by connecting more intelligent technologies, and the society that uses them to make more precise decisions. (IIoT: An analysis framework, n.d.)

4 EXISTING ISSUES OF IOT

4.1 Privacy

The Internet of Things has various advantages for the consumers and is cable of revolutionising the way users interact with technology. The most likely scenario of development of the IoT is a combination of the digital and

physical realities in a manner that is difficult to fathom at the moment. However, we know that it will require a use of various sensors and devices that collect a lot of personal information about the users. Most of such devices will be and are found already in private places such as all around the home, in the car, and even on the body of the user. All of these devices will read and collect a lot of private information about their owners and the main threat and challenge is to keep it all secure and protected. (Standards problem in IoT, n.d.)

This is not just a potential problem that might occur in the future, it is an issue that is serious even today because people are already sharing and providing their personal data to the devices. Nowadays, most smartphones, wearables and other gadgets are tracking owner's location and older technology that does not do that is becoming obsolete. It is obvious that the majority of consumers do not realize that information about their new devices track their movements. (Standards problem in IoT, n.d.)

There have been multiple cases relating to the problems of consumer privacy, and the most recent one involved Samsung Smart TV. In its privacy policy it suggested to avoid discussing intimate conversations near the gadget, which the public compared to a similar situation from George Orwell's book 1984. This resulted in a lot of criticism against Samsung and forced them to change the way the Smart TV reads and collects user data. (Standards problem in IoT, n.d.)

Another problem is that most consumers do not pay too much attention to the terms and conditions of use for every technology or application they have and in case an average person tried to read privacy policies, they would most likely not be able to understand it due to its legal language. In addition to that, most devices come with obligatory arbitration clauses that leave no room for consumers to sue the company if their product harmed them. As an outcome, there is no guarantee of consumer privacy and there are no simple ways to legally punish the company for causing harm to the customers. (Reasons People Aren't Embracing, n.d.)

A main step for solving the issue of privacy in the IoT is improving corporate transparency. Governments could introduce laws that demand companies to collect real and provable consent from customers before collecting private user data. Customers should have an ability to see what data is gathered, where it is stored and how it is utilised.

In reality, it is in tech company's interests to provide maximum privacy to their customers. It has been the case with the Alliance of Automobile Manufacturers, when there was a study about their customers being concerned about the privacy and security of data of smart vehicles. As a result, the manufacturers introduced new privacy terms and settled with the demand.

Smaller business can follow the examples and apply the same privacy policies of the larger ones. Those companies that cannot ensure complete data security should not collect any private user data at all. (Reasons People Aren't Embracing, n.d.)

An example of a successful implementation of privacy policies comes from a company named FitBit. As most of the IoT companies trade in a free market, they have an opportunity to decide on their own specifications of standards and regulations. (Reasons People Aren't Embracing, n.d.)

The most commonly used type of privacy policies by the most IoT corporations is called layered privacy. It consists of three different layers: the legal code, the human-readable layer and the machine-readable layer.

Law makers would write legal code policies and pass them on to the judges for interpretation. There should be a simplified and shortened version of the privacy policy for the average customer to read and understand it. This can be called a layer readable by humans. On the other side, there should be a layer readable by machines. This layer would contain only the information that is accessible and permitted by the user and it would be written in a way more understandable by search engines, programmes and different types of technology. (Reasons People Aren't Embracing, n.d.)

All of these projects have the potential to revolutionise the way consumer data is protected, but these methods are not perfect. Corporations should have legal restrictions that would not let them break the terms and conditions with their users. However, in reality companies are able to win any lawsuit due to unfair clauses that are hidden among the terms of use and are often printed in a tiny font size. (Standards problem in IoT, n.d.)

The Consumer Financial Protection Bureau has concluded that clauses of arbitration barriers on activities of class continuously damage the interests of the society. The main reason for that is because such lawsuits shine light into obscure activities of corporations and without such publicity, society would never know of these practices. A solution has been proposed by the agency to prohibit obligatory arbitration clauses for the majority of products and services.

A surprising place where this has been a problem is schools, which is why several governments are prohibiting any pre-supposed arbitration clauses for private schools. This give an opportunity to sue the school in case a student was abused. The same rules concerning lawsuits should be applied for companies that trade the IoT products and software. (Standards problem in IoT, n.d.)

This is a very complex issue because it is related to customer privacy and it affects multiple industries, companies, customers, global community and

a proper solution would require a discussion that includes all of the parties. The priority is to satisfy and protect consumers privacy and it is a responsibility of corporations and industries to do that. (Security and Privacy Issues, n.d.)

There is more and more electronic smart the Internet of Things devices available in the market. An insignificant security problem on smartphone can evolve into massive and serious issues when taking into account different IoT devices in a smart home. In order to understand the magnitude of the security issue, it is crucial to recognize the way IoT devices function. (Security and Privacy Issues, n.d.)

Due to IoT devices having few assets, it is impossible to apply common security frameworks. For this reason, either unique security suits have to be created or non-standard ones should be utilized. It is important to note that only practical and realistic protection options have be considered.

The way IoT devices are designed is closely related and directly affects privacy and security of data. Privacy must ensure solid protection of user's data from unauthorized access. (Security, Privacy and Safety, n.d.) Every single user has the rights to privacy. However, the fact that users have this right, it does not prevent them from being tracked and it does not protect their data in practice. Almost any user activity leaves certain types of traces on the internet. There are many cases of giant corporations and governments collecting these sorts of private user data and taking advantage of it without society's consent. (Security and Privacy Issues, n.d.)

It is not uncommon for hackers and companies to exploit user privacy. According to a German research in data encryption, it is possible to intercept signals from smart home devices like a smart TV and to see what television channel a person is watching. Unless these issues are addressed, the progress in development of the IoT will be slowed downed drastically.

4.2 **Security**

While the rise of the IoT in the modern workplace may be novel, several of the most common security weaknesses plaguing the IoT devices aren't exactly new.

The common IoT vulnerabilities that are often seen tie into poor security on mobile applications. One of the biggest problem is data being stored on mobile applications. When saving data on the iOS is likely less risky than on the Android, storing private data on any mobile device is less than perfect. What happens if a worker simply loses a phone with private data on it that isn't backed up elsewhere? (Security, Privacy and Safety, n.d.)

The cloud is often another weak link in the IoT implementations. Cloud APIs for the IoT devices are probably worse than normal web APIs when it comes to security. Part of the problem is that most of the developers look at the communications between the IoT devices and the cloud APIs as machine-to-machine communication. (Security, Privacy and Safety, n.d.)

And then there's the risk that the IoT devices have not-so-secret backdoors that can be accessed using default usernames and passwords. Companies should be asking: 'Are we doing proper authentication and encryption of our communications?'. Users could have someone Telnet onto a nonstandard port and use a default password. (Security, Privacy and Safety, n.d.)

Another the IoT security concern is that common devices such as smart TVs and printers can become threat vectors. There are potentially significant security issues around multifunction printers. (Standards problem in IoT, n.d.)

A lot of companies don't have the equivalent of endpoint security for every connected device whether they are printers or security cameras or something else. How do you know if one of these devices is doing something it shouldn't be doing? (Standards problem in IoT, n.d.)

Yet another the IoT security concern is the growing prevalence of rogue connected devices hidden within enterprises secretly surveilling the network. There are rogue IoT devices, for instance, Raspberry Pi or WiFi Pineapple devices. This is a popular attack vector because it is so easy. An attacker can take one of these devices and go to Midtown Manhattan of NYC and make hundreds of devices connect to a rogue device — including mobile devices belonging to financial institutions and other types of companies. (Security, Privacy and Safety, n.d.)

A related problem is that many organizations aren't fully aware of what is on their network and thus can't judge if they have misconfigured IoT devices or rogue devices. It is often difficult to maintain a dashboard-like view of all the devices on the network. (Security, Privacy and Safety, n.d.)

Ultimately, the field of the IoT security is a multifaceted, far-ranging discipline. The entire ecosystem, which includes all the above factors as well as considerations like vulnerabilities posed by specific radio frequency communications. Bluetooth 5, for instance, supports mesh networking, potentially enabling an attacker to target a single Bluetooth device and spread malware across the entire mesh network. (Security, Privacy and Safety, n.d.)

Manufacturers of the IoT devices guarding intellectual property may want to consider protecting the firmware of their products. For companies

where IP is important, they don't want someone easily getting the firmware off the device.
Many manufacturers of the IoT devices take a lackadaisical approach to protecting intellectual property, failing to use cryptography-based protection that is available on the hardware they use. There are oftentimes organizations don't use built-in protection that exists on some of their chipsets. (Security, Privacy and Safety, n.d.)

As a growing number of security researchers are beginning to focus on the Internet of Things, some of their findings pose more of a theoretical risk than an actual one — for now. There are some pieces on the concept of using smart lighting to exfiltrate data by compromising an internal network. Is it possible to get a light in a room to fluctuate enough to transmit data? In the test environment, it is very doable. In the real world, it has still not been proven yet — at least that currently known. (Security, Privacy and Safety, n.d.)

4.3 **Standards**

The current situation of interoperability levels is an obvious issue. It is in nature a very complicated challenge because it is not clear who should enforce universal standards and what should they be. Furthermore, the IoT industry develops at such high rates that they may constantly require new types of standards and make the old ones irrelevant. Some of the common standards are the Linux-backed AllJoyn, Intel's Open Interconnect Consortium, IEEE P2413, and the ITU-T SG20 standard for smart cities. A recent innovation in the field of standards of IoT security is UL 2900 certification. (Standards problem in IoT, n.d.)

Governments of Japan and Germany have recently decided to introduce the IoT security standards for the technology market and industry. Their standard is based on the Thread IPv6 which is applicable for networks, data security, use of power and interoperability. (Standards problem in IoT, n.d.)

As long there are not clearly defined security standards, the whole IoT industry is endangered. However, there are examples of successful security standards that have been working for years. The main ones include wireless protocols such as Wi-Fi, Bluetooth and ZigBee. These ones have been tested by time and billions of users and have proven to be secure and stable. Therefore, there is a need of security standards for the newly created and introduced IoT devices. It all comes down to the issue of deciding on some sort of universal standards, but it is complicated because it might result in fragmenting the IoT industry. (Standards problem in IoT, n.d.)

To explain this point better, let us consider as an example a situation where there were three distinct types of Wi-Fi connection. When a customer wishes to purchase a device with Wi-Fi connectivity, a right type of connection would have to be selected. However, this creates a problem where a user is limited to using only specific type of connection and that may not be available everywhere. In order to avoid it and solve this issue, someone would have to come up with universal standards for Wi-Fi connectivity, that would make things a lot more efficient, and this is what we have in the current world.

This sort of situation is very common with unlicensed IoT such as Sigfox, LoRa, Ingenu, Telensa, Weightless, etc. Most of these devices are conflicting with one another, even though they function in a similar range. This is pointless and inefficient in every way possible. (Standards problem in IoT, n.d.)

Manufacturers explain it by saying that their technologies have different target audiences – for instance, LoRa is utilized by complex devices, while Sigfox targets less expensive devices. However, such explanation does not consider that a universal standard would satisfy both market segments, as in the same manner Wi-Fi or cellular are used by various types of devices. There is the same cellular technology for low-bandwidth users and for high-bandwidth users. Having such universal standard is beneficial from an economic point of view because it benefits from economies of scale and it inspires global deployment of infrastructure. (Standards problem in IoT, n.d.) On the other side, some claim that competition at early stages of development of IoT is actually advantageous because it pushes the progress forward. (Standards problem in IoT, n.d.)

5 EMPIRICAL PART OF PROJECT

Apinf is a web-based application written mainly in JavaScript and JavaScript based libraries. In order to start writing API, it was determined what function and data the API will imitate.

The task was to get some data about current status of controlled and managed by Apinf APIs. In other words, Apinf hosts APIs added by users, one of the features of the Apinf platform is the ability to see the current status and history of API availability. This functionality had to be extended and improved. For the existing and created functionality the author of this report was asked to create a basic API to retrieve data. The empirical part of this report includes new UI elements and a method of acquiring data from the database through an API. The software application code can be found in Appendices 2 to 6.

5.1 **Apinf and API Management Platform**

Apinf API Management Platform is software that allows the user to have more control and functionality over his/her APIs. Apinf is used by companies with existing API functionality. Apinf simply hosts an API specification and allows to share it through a so-called API profile which simply is a page with a constant address. This API profile allows companies to share information about theirs API(s). Apinf plays an important role as a part of the IoT chain of applications, allowing companies to share more data with their users and developers. This role is mostly the visualisation of API usage, through charts, user management and API cataloguing. Apinf consists of two parts: a proxy and a client-side platform.

The proxy part connects API calls and makes them available for analytics, rate limiting, user management and in general it allows for more control than traditional direct REST calls.

The client-side part is a web-based platform that gives the user access to information exposed by a proxy. The platform consists of different views, each dedicated to a certain part of API management. Users have the ability to set up the API profile page to add their API to a public catalogue, the can also configure a sharable link, access proxy configuration, add outside contribution using email, upload and edit an API specification to help developers understand the technical aspects of API and see who is using it. There is also a possibility to enable API monitoring feature. It allows to see the current status of the API backend, which is very important when it comes to the determination the health of API.

Apinf uses the API key to determine and identify users. This is a secure method which allows privately connect users using a proxy to their call analytics. These analytics are displayed using charts and tables, which help to discover errors, slow responses and measure traffic which goes through the proxy of API. The API key also allows to set limits, and in general to manage users.

5.1.1 Overview

Apinf is an open-source software developed with open-process principals. It is hosted via Github and available for outside contributions. Apinf can be used through the SaaS web-application publicly, and it also can be deployed as a dedicated application.

Although the proxy is one of the key components of Apinf, it has been developed and supported fully by the outside community. Apinf has been adopting a proxy component to fulfil its needs. With this in mind, the overview part of this thesis covers only the client-side part of the Apinf

platform. Th main component of Apinf is a dashboard illustrated in Figure 5.

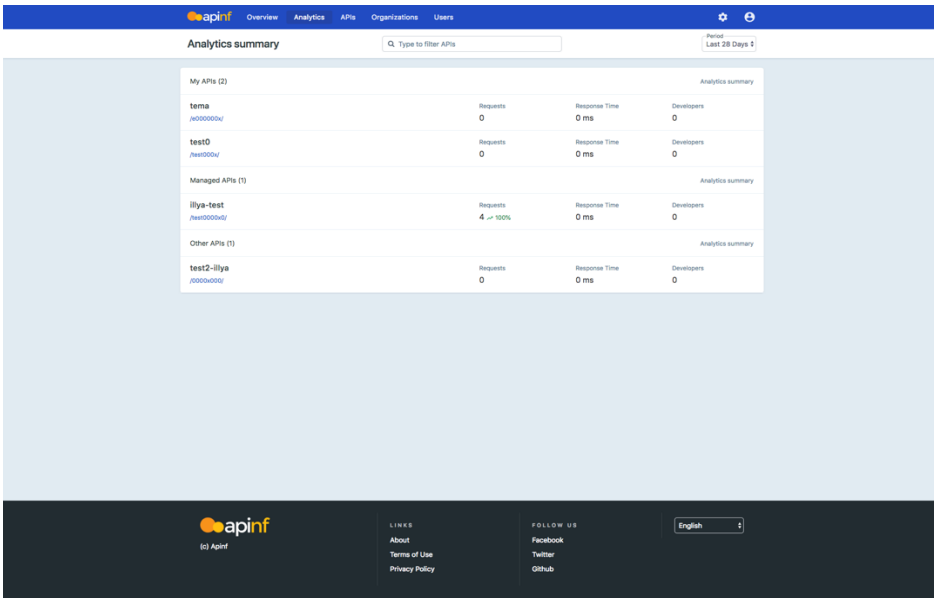

Figure 5. Analytics dashboard

The dashboard represents API analytics. Analytics consist of a number of requests made through the selected proxy, the response time and the number of developers who have made these requests. This information helps the owners of API to understand how and who is using their APIs.

Another important component is the API catalogue as shown in Figure 6. It serves two purposes: it shows users created and added APIs and creates a way to discover and advertise open APIs to users outside of Apinf.

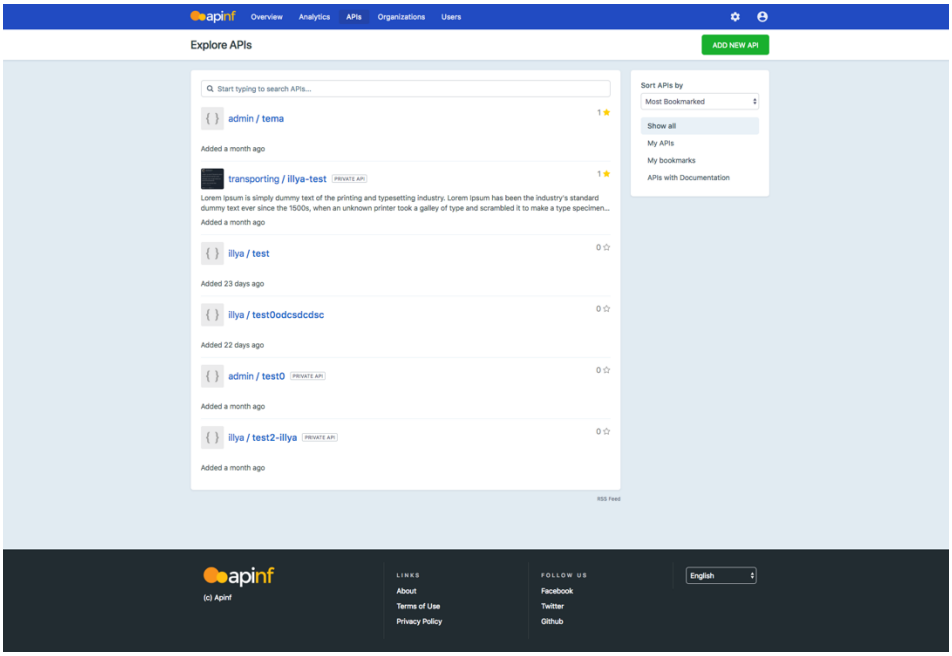

Figure 6. API catalogue

An API catalogue consists of a list of APIs with a picture, a brief description and a name. It is possible to find APIs using the search field. The API catalogue allows one to sort and filter APIs.

The API profile (Figure 7) is the key component of an API configuration. It consists of several views depending on selected preferences. The main idea of an API profile is to give users an understanding of the API content and how to use it. It is up to the owner of the API to enable and setup each additional component of the API profile to help with technical aspects of API. In Figure 7 are listed examples of API profile components.

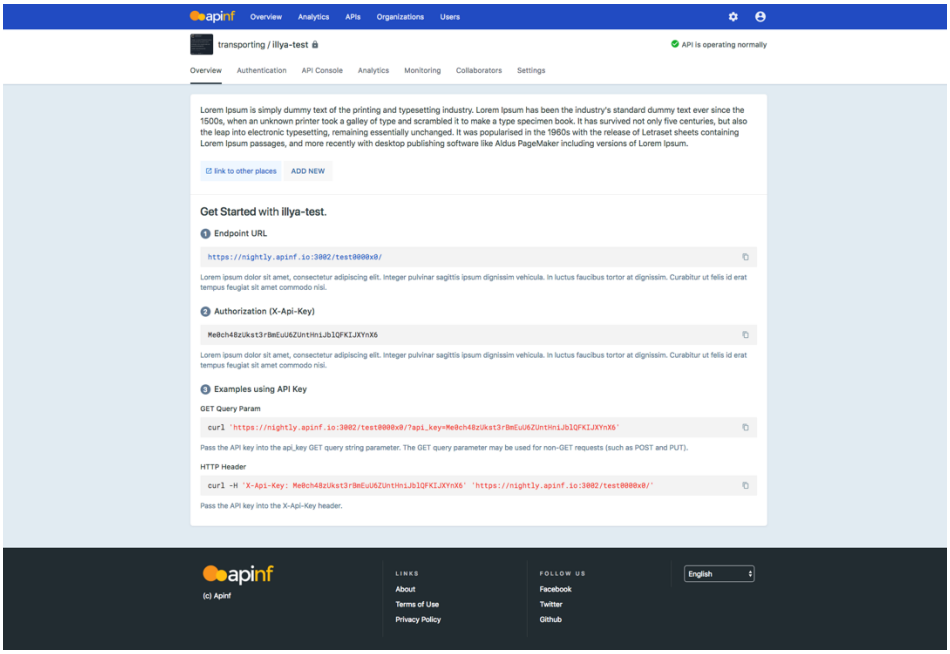

Figure 7. API profile

An API console illustrated in Figure 8 plays a significate role in assisting users with understanding API usage. Apinf implements this with Swagger documentation technologies, which break every type of requests into different categories. The UI includes numerous elements: descriptions, try-out methods, models and authentication.

The Swagger documentation can be uploaded via specifically formatted .json files or directly via URL. Options also allow setting several additional parameters.

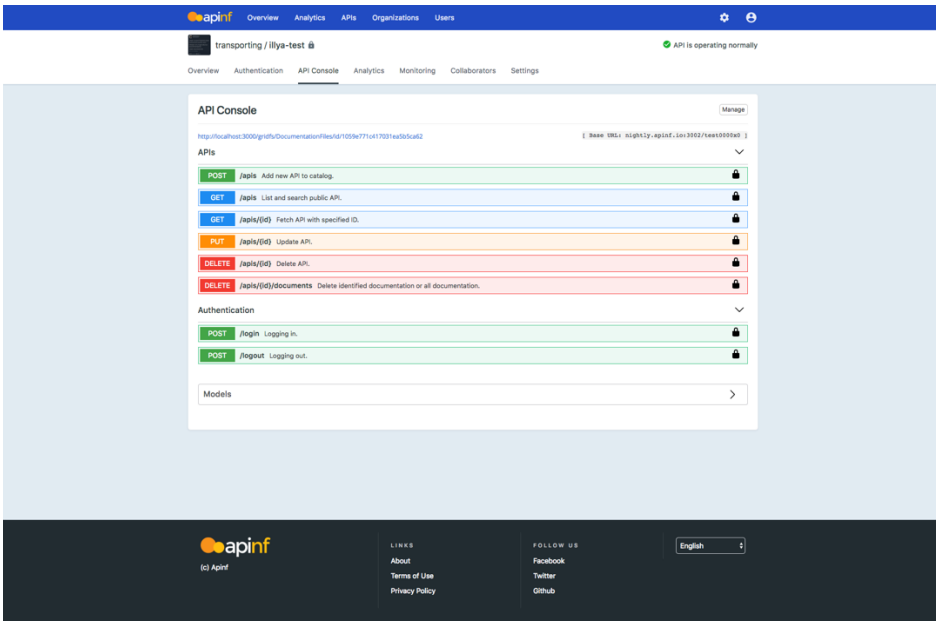

Figure 8. API console with specification

Apinf allows to manage and control users, for example allowing only certain developers to access and edit the API configuration as seen in Figure 9.

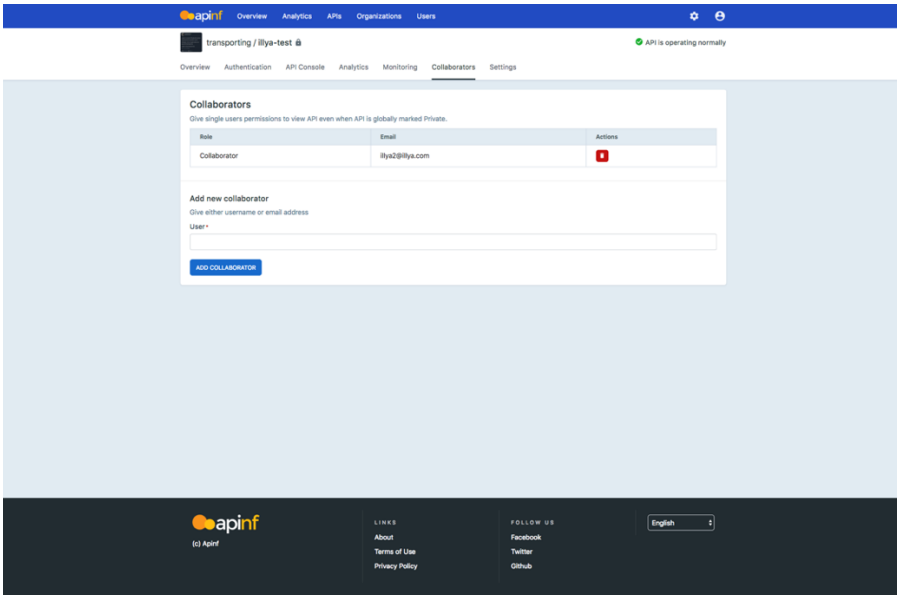

Figure 9. Collaborators page of API profile

Apinf allows the API owners to manage an API in different ways, including branding, organizational or networks settings, as shown in Figure 10.

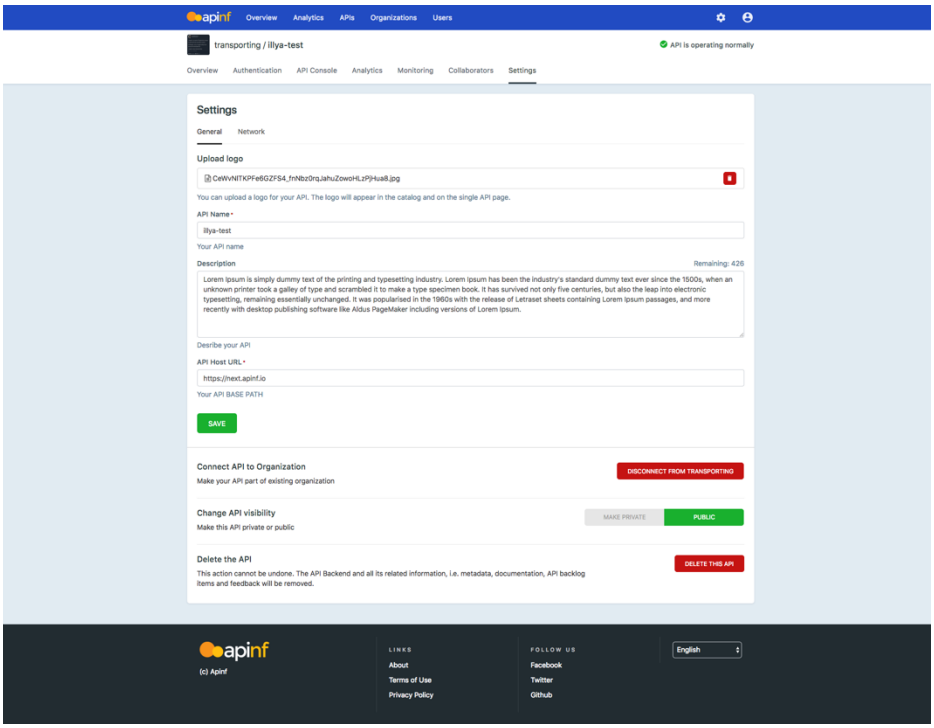

Figure 10. Settings of API profile page

5.1.2 Software architecture

Apinf is built using numerous technologies, such as API Umbrella, Bootstrap, Clipboard.js Crossfilter, d3.js, dc.js, Intro.js, Lodash, Meteor.js Moment.js, Mongo DB, Simple Statistics, Swagger.js, Swagger UI, URI.js. Key principal of Apinf's stack of technologies is to use and contribute to existing open-source projects.

5.2 **API Development**

The API availability functionality, also known as the API monitoring consist of server response code and short explanation text along with date and time the request was made. This information helps user to understand if API is working on not. Basic functionality has been already implemented by company contributors.

During this research additional functionality was created, which includes: expanding user-interface with a historical data table, icons to visually enhance accessibility of information, and time stamps of requests. In addition to user-interface changes, most of the server-side code was rewritten for efficiency. Once improvements of application itself were approved by accepting party, work on API to expand the API monitoring functionality has been started.

API monitoring allows users to check on API availability. Apinf automatically calls selected endpoint and returns status code. In order to enable API monitoring user is required to input endpoint URL and save it (Figure 11 and 12). The endpoint is being saved in database and with first initial call result (Appendix 1). Every hour application will make a call to selected endpoint URL and store result in database.

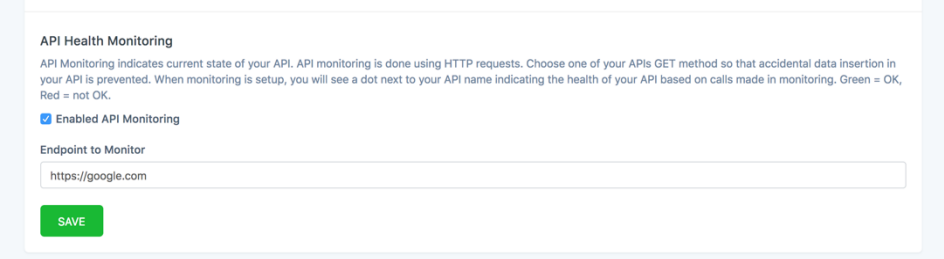

Figure 11. Component to enable API monitoring

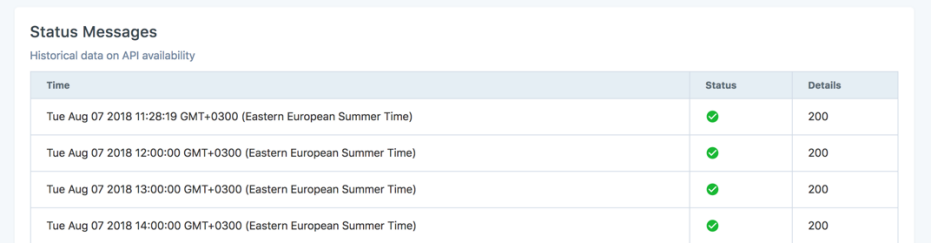

Figure 12. Table with historical results of API availability

The API definition process starts with defining new path and route. Defined route will be part of API documentation and will be used to access any further functionality and features and API. For the purposes of monitoring API */monitoring* route was created,

```
CatalogV1.addRoute('apis/:id/monitoring'…
```

every request related to using API for the purposes of monitoring data, such as getting, posting or modifying must be made at this route. Full code can be found in Appendix 7 through 9.

Next step is related to authentication. Every API can be restricted to users with specific authorization rights. In case of monitoring API part authentication is not required, which means that some data can be accessed anonymously.

Every new part of APIs must be well documented. Apinf is using Swagger documentation for these purposes. Swagger allows to break API by paths and requests within Swagger UI. In order to make new path part of existing Swagger documentation tag specific to this API was added.

```
CatalogV1.swagger.tags.api
```

Plain language describing functionality helps developers use API more effectively and efficiently. Swagger allows to add these descriptions inside of Swagger UI (Figure 13).

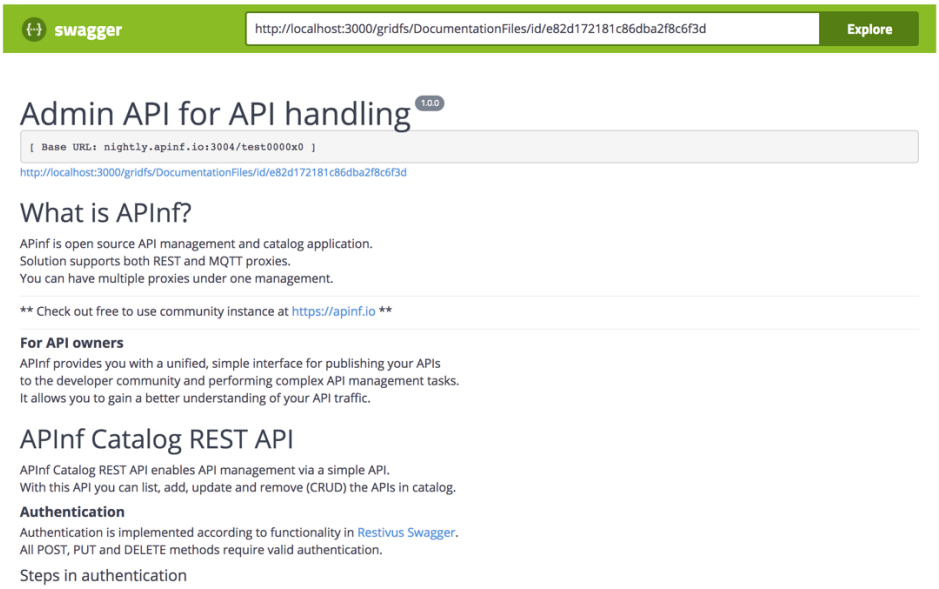

Figure 13. Example of Swagger documentation with description

Responses are defined by HTTP responses server numbers. Implementation of monitoring API consists of several such numbers. (Table 1)

200	Success: Latest monitoring status of API
400	Error: Bad Request. Erroneous or missing parameter.
401	Error: Authentication is required
403	Error: User does not have permission
404	Error: API is not found

Table 1. Server response numbers

Core features of monitoring API are following application functionality. Functionality of API is described in *action* part.

```
action () {}
```

First of part of *action* defines application data sources. Since monitoring is part of API, related data collection is being selected.

```
const api = Apis.findOne(apiId);
```

Next steps are API creation are determining numerous cases of possible behaviour. Making sure that user gets the correct data according to request and permissions.

After all data collection and checking are done API forms an answer.

```
return {
      statusCode: 200,
      body: {
        status: 'success',
        data: apiMonitoringResponse,
      },
    };
```

The main focus was on how to improve overall experience of using Apinf platform, not just technical, code-based contributions. As a first step, it was researched what data users might be interested in, while using this API. It was determined that both accepting party and users, of Apinf platform are interested in getting latest code status with all related meta information.

```
{
    "status": "success",
    "data": {
        "_id": "72NhsGqSCaZ4ZXti6",
        "enabled": true,
        "responses": {
            "date": "2018-09-12T10:33:09.477Z",
            "server_status_code": "404"
        }
    }
}
```

Secondly, it was researched how other parts of existing API implementations are working and what are their technical specifications. It is important that all new additions to code base of application are done in same manner and following the same code style. Once all preparations and discussions with developers of Aping were done, process of practical implementation of monitoring API has begun.

6 SUMMARY

Aims of this thesis project were:

- Describing the current state of the Internet of Things and describing the empirical part of this topic by exploring technical aspects through APIs
- Research examples of implementations of the IoT, with their purposes and technical aspects.
- Addressing current and future issues of IoT
- Creating monitoring API for Apinf API Management Platform

The aims were successfully achived and approved by the commissioned part. The proposed changes to monitoring API were merged and now are a part of the Apinf API Management Platform.

The Internet of Things being a relatively new term has already influenced and shaped numerous areas of our society. The IoT became part of not only industrial and professional areas, such as healthcare, agriculture, cities, cars but also the everyday life of people. Concepts and ideas of IoT are already deeply rooted into many industries, things simple as watches and home appliances are not only connected to the Internet, but also are able to communicate with each other.

The potential of the IoT is so vast that cannot be properly described and measured. The reason for this is not only its potential in application, but also in possible issues and work on solving these issues. This means that the area of IoT influence is so big, people now are not able to properly understand and comprehend its consequences. This includes: security problems, privacy issues, the impact on people work places, etc.

Part of this research was to examine empirically how to improve communication with the IoT systems. The API of any software or hardware system is a crucial area of communication between the user and the software, the hardware or even the software and the hardware itself. As language is the most efficient way of human communication, the API is most efficient way of the IoT commutation. In a nutshell the API is any number of commands and rules designed to access certain part of the software or hardware to get data to control them. During this project, the Apinf web-based software was taken as an example of the modern IoT system. The purpose of the Apinf is to provide users with an ability of managing API(s), including analytics, user-management and cataloguing.

The aim of this research project was to prove that the IoT system requires a well written API. The API is not just a part of modern application development, it is a crucial part of a communicational language and core

idea of a world connected by the Internet. There should be no system in the world without a proper method of interaction with it.

As an example, to prove this idea, one part of the Apinf system was chosen to write an API. The created API helps users to interact with the API monitoring feature of the Apinf. It made this part of the Apinf more user-friendly, opened a possibility for the developers to get data and control it as well as gave potential to reuse functionality in other places. The system became more connected with other parts of the Internet.

REFERENCES

A Brief History of the Internet of Things, (n.d.) Retrieved March 2, 2018, from http://www.dataversity.net/brief-history-internet-things/

A guide to healthcare IoT possibilities and obstacles. (n.d.) Retrieved March 10, 2018, from https://searchhealthit.techtarget.com/essentialguide/A-guide-to-healthcare-IoT-possibilities-and-obstacles

Connected Car Architecture and Virtualization. (n.d.) Retrieved April 22, 2018, from https://www.researchgate.net/publication/301272903_Connected_Car_Architecture_and_Virtualization

Connected Vehicles: Solutions and Challenges. (n.d.) Retrieved April 24, 2018, from https://www.researchgate.net/publication/264564008_Connected_Vehicles_Solutions_and_Challenges

Future connected vehicles: challenges and opportunities for spatio-temporal computing. (n.d.) Retrieved April 20, 2018, from https://www.researchgate.net/publication/311491335_Future_connected_vehicles_challenges_and_opportunities_for_spatio-temporal_computing

Health Monitoring and Management Using Internet-of-Things (IoT) Sensing with Cloud-based Processing: Opportunities and Challenges. (n.d.) Retrieved March 11, 2018, from http://www2.ece.rochester.edu/~gsharma/papers/Moeen_HealthMonitor_SCC2015.pdf

How APIs Are Driving Smart Cities (n.d.) Retrieved October 9, 2018, from https://nordicapis.com/how-apis-are-driving-smart-cities/

Internet of Things (IoT): Security, Privacy and Safety. (n.d.) Retrieved March 5, 2018, from https://datafloq.com/read/internet-of-things-iot-security-privacy-safety/948

industrial internet of things (IIoT). (n.d.) Retrieved March 20, 2018, from https://internetofthingsagenda.techtarget.com/definition/Industrial-Internet-of-Things-IIoT

Security and Privacy Issues in IoT. (n.d.) Retrieved March 4, 2018, from https://techcrunch.com/2016/08/14/the-iot-threat-to-privacy/?guccounter=1

Smart cities. (n.d.) Retrieved March 26, 2018, from http://folkestone.com.au/wp-content/uploads/2016/07/Smart-Cities-The-Future-of-Cities-February-2016.pdf

Smart city. (n.d.) Retrieved March 18, 2018, from https://internetofthingsagenda.techtarget.com/definition/smart-city

Smart Home. (n.d.) Retrieved May 6, 2018, from https://www.investopedia.com/terms/s/smart-home.asp

The Architecture of Wearable Technology. (n.d.) Retrieved April 26, 2018, from https://www.researchgate.net/publication/278327463_The_Architecture_of_Wearable_Technology

The industrial internet of things (IIoT): An analysis framework. (n.d.) Retrieved March 28, 2018, from https://www.sciencedirect.com/science/article/pii/S0166361517307285

The Internet of Everything for Cities. (n.d.) Retrieved March 28, 2018, from https://www.cisco.com/c/dam/en_us/solutions/industries/docs/gov/everything-for-cities.pdf

The possibilities of Internet of Things (IoT) for Agriculture. (n.d.) Retrieved March 13, 2018, from http://www.fao.org/e-agriculture/news/possibilities-internet-things-iot-agriculture

There's a standards problem in IoT, here's how to solve. (n.d.) Retrieved March 1, 2018, from https://internetofbusiness.com/standards-iot-problem-sigfox/

There's a standards problem in IoT, here's how to solve. (n.d.) Retrieved March 7, 2018, from http://www.ijcnis.org/index.php/ijcnis/article/view/2074/193

Three big challenges for smart cities and how to solve them. (n.d.) Retrieved March 18, 2018, from https://theconversation.com/three-big-challenges-for-smart-cities-and-how-to-solve-them-59191

Top 10 Reasons People Aren't Embracing the IoT. (n.d.) Retrieved March 4, 2018, from https://www.iotworldtoday.com/2016/04/20/top-10-reasons-people-aren-t-embracing-iot/

What is a 'smart city' and how it will work. (n.d.) Retrieved March 26, 2018, from http://www.ijcnis.org/index.php/ijcnis/article/view/2074/193

What is an IoT Platform?, (n.d.) Retrieved March 2, 2018, from https://www.iotforall.com/what-is-an-iot-platform/

What is Industry 4.0—the Industrial Internet of Things (IIoT)?, (n.d.) Retrieved September 18, 2018, from https://www.epicor.com/resources/articles/what-is-industry-4-0.aspx

When the Internet of Things becomes the Internet of Stings, (n.d.) Retrieved March 2, 2018, from https://www.netnames.com/insights/blog/2017/01/when-the-internet-of-things-becomes-the-internet-of-stings/

Why IoT, big data & smart farming are the future of agriculture. (n.d.) Retrieved March 16, 2018, from https://www.businessinsider.com/internet-of-things-smart-agriculture-2016-10?r=US&IR=T&IR=T

5 IoT applications retailers are using today. (n.d.) Retrieved March 12, 2018, from https://www.sas.com/en_us/insights/articles/big-data/five-iot-applications-retailers-are-using-today.html

Appendix 1

Content of monitoring.html

```
<template name="apiMonitoring">
 {{# if apiMonitoringSettings.enabled }}
  <h2 style="font-size:1.4em;margin-top:0;">
    Status Messages
  </h2>
  <p style="color:#6d859e;">
    Historical data on API availability
  </p>
  <table class="rtable">
    <thead>
      <tr>
        <th>Time</th>
        <th>Status</th>
        <th>Details</th>
      </tr>
    </thead>
    <tbody>
      {{# each apiStatusData }}
      <tr>
        <td>{{ this.date }}</td>
        <td>
          {{# if apiStatusCode (this.server_status_code) }}
          <i class="mdi mdi-checkbox-marked-circle mdi-18px
status-success"></i>
          {{ else }}
          <i class="mdi mdi-alert-circle mdi-18px status-
warning"></i>
          {{/ if }}
        </td>
        <td>{{ this.server_status_code }}</td>
      </tr>
      {{/ each }}
    </tbody>
  </table>

  <hr style="color: rgba(59, 59, 88, 0.15); margin-right: -
20px; margin-left: -20px; margin-top: 2em; margin-bottom:
2em;">

  {{/ if }}

  <h3 style="font-size:16px; margin-top:0;">
    {{_ "apiMonitoring_panelTitle_Monitoring" }}
  </h3>
  <p                              style="color:#6d859e;">{{_
"apiMonitoring_helpIcon_text" }}</p>
  <div class="form-group" style="margin-bottom:0;">
    {{#       autoForm       collection=monitoringCollection
id="apiMonitoringForm"                          type=formType
doc=apiMonitoringSettings }}
    <!-- hidden field, auto-value -->
    {{>     afQuickField     name='apiId'     value=api._id
type="hidden" }}
    {{> afQuickField name='enabled' }}
    {{# if afFieldValueIs name='enabled' value=true }}
    <label for="endpoint-monitor-field">
      {{ afFieldLabelText name='url' }}
```

```
    </label>
    {{> afFieldInput name='url' }}
    {{/ if }}
    <button type="submit" class="btn btn-success btn-
success-special" id="save-monitoring-settings">
      {{_ "apiMonitoring_saveButton_text" }}
    </button>
    {{/ autoForm }}
  </div>
</template>
```

Appendix 2

Content of monitoring.js

```
// Meteor packages imports
import { Template } from 'meteor/templating';

// Collection imports
import { MonitoringSettings, MonitoringData } from
'/apinf_packages/monitoring/collection';

// APInf import
import convertStatusCode from
'/apinf_packages/apis/client/profile/status/convert_status_
code';

Template.apiMonitoring.onCreated(function () {
  // Get reference of template instance
  const instance = this;

  // Get api id
  const apiId = instance.data.api._id;

  // Subscribe on Monitoring collection
  instance.subscribe('monitoringSettings', apiId);
  instance.subscribe('getApiStatusRecordData', apiId);
});
Template.apiMonitoring.onRendered(() => {
  // Show a small popup on clicking the help icon
  $('[data-toggle="popover"]').popover();

  // Init tooltip
  $('[data-toggle="tooltip"]').tooltip();
});

Template.apiMonitoring.helpers({
  apiMonitoringSettings () {
    // Get api id
    const apiId = this.api._id;

    // Get api monitoring document
    return MonitoringSettings.findOne({ apiId });
  },
  monitoringCollection () {
    // Collection for autoform
    return MonitoringSettings;
  },
  formType () {
    // Get API ID
    const apiId = this.api._id;
```

```
    // Look for existing monitoring document for this API
    const existingSettings = MonitoringSettings.findOne({
apiId });
    if (existingSettings) {
      return 'update';
    }

    return 'insert';
  },
  apiStatusData () {
    const apiId = this.api._id;
    const monitoringData = MonitoringData.findOne({ apiId });
    if (monitoringData) {
      return monitoringData.responses;
    }
    return [];
  },
  apiStatusCode (code) {
    return code === '200';
  },
});
```

Appendix 3

Content of schema.js

```
// Meteor packages imports
import { SimpleSchema } from 'meteor/aldeed:simple-schema';

// Collection imports
import { MonitoringSettings, MonitoringData } from './';

// Describe collection for store data associate with
monitoring settings
MonitoringSettings.schema = new SimpleSchema({
  apiId: {
    type: String,
  },
  enabled: {
    type: Boolean,
    optional: true,
  },
  data: {
    type: String,
    optional: true,
  },
  url: {
    type: String,
    regEx: SimpleSchema.RegEx.Url,
    custom () {
      const monitoringUrlEnabled =
this.field('enabled').value;
      const monitoringUrl = this.value;
      let validation;

      if (monitoringUrlEnabled === true && !monitoringUrl) {
        validation = 'required';
      }
      return validation;
    },
  },
});
```

```
// Describe collection for store data form server
MonitoringData.schema = new SimpleSchema({
  apiId: {
    type: String,
  },
  responses: {
    type: [Object],
    optional: true,
  },
  'responses.$.date': {
    type: Date,
    optional: true,
  },
  'responses.$.server_status_code': {
    type: String,
    optional: true,
  },
});
// Enable translations (i18n)
MonitoringSettings.schema.i18n('schemas.monitoring');

MonitoringSettings.attachSchema(MonitoringSettings.schema);
MonitoringData.attachSchema(MonitoringData.schema);
```

Appendix 4

Content of status.html

```
<template name="viewApiStatus">
  <div class="pull-right api-status">
    <i class="mdi {{ statusIcon }} {{ classList }}" data-
toggle="tooltip"     data-placement="top"     data-original-
title="{{ originalTitle }}" style="font-size: 1.8rem;"></i>
    <span     class="status-message"     style="font-weight:
500;color: #55626c;">{{ originalTitle }}</span>
  </div>
</template>
```

Appendix 5

Content of status.js

```
// Meteor packages imports
import { Template } from 'meteor/templating';

// APInf import
import convertStatusCode from './convert_status_code';

Template.viewApiStatus.onRendered(() => {
  // Init tooltip
  $('[data-toggle="tooltip"]').tooltip();
});

Template.viewApiStatus.helpers({
  classList () {
    // Get api
    const api = Template.currentData().api;
    // Get class name depending on the api status code
    const          {          className          }          =
convertStatusCode(api.latestMonitoringStatusCode);
```

```
    // Create a new line using join
    return [
      `api-status-indicator-${api._id}`,
      className,
    ].join(' ');
  },
  originalTitle () {
    // Get api
    const api = Template.currentData().api;

    // Get original title depending on the api status code
    const { statusText } =
convertStatusCode(api.latestMonitoringStatusCode);

    return statusText;
  },
  statusIcon () {
    // Get api
    const api = Template.currentData().api;

    const { statusIcon } =
convertStatusCode(api.latestMonitoringStatusCode);

    return statusIcon;
  },
});
```

Appendix 6

Content of status.less

```
/* Bootstrap */
@import "/apinf_packages/core/client/style/bootstrap-
variables.less";

.icon-indicator {
  display: inline-block;
  border-radius: 50%;
  -moz-border-radius: 50%;
  -webkit-border-radius: 50%;
  margin-bottom: 0.1em;
}

.api-status {
  position: absolute;
  right: 15px;
  top: 16px;
}

.status-wait {
  color: #a0a0a0;
}

.status-success {
  color: @brand-success;
}

.status-warning {
  color: @brand-warning;
}

.status-danger {
  color: @brand-danger;
}
```

```
.status-info {
  color: @brand-info;
}

@media only screen and (max-width: 768px) {
  span.status-message {
    display: none;
  }
  .api-status {
    right: 0;
  }
}
```

Appendix 7

Content of convert_status_code.js

```
// Meteor packages imports
import { TAPi18n } from 'meteor/tap:i18n';

// Check which status code is received
// and set class name and status text depending on it
export default function convertStatusCode (serverStatusCode)
{
  // Init variables
  let className = '';
  let statusText = '';
  let statusIcon = '';

  // Computed an api status for Switch
  const apiStatus = Math.floor(serverStatusCode / 100);

  switch (apiStatus) {
    //  Temporary  status:  monitoring  is  enabled  but  real
status code is unknown
    case 0:
      className = 'status-wait';
      statusText                                              =
TAPi18n.__('viewApiStatus_statusMessage_Loading');
      break;
    // Informational status code
    case 1:
      className = 'status-info';
      statusText = `

${TAPi18n.__('viewApiStatus_statusMessage_ErrorCodeText')}
          ${serverStatusCode}.

${TAPi18n.__('viewApiStatus_statusMessage_Informational')}
          `;
      break;
    // Success status code
    case 2:
      className = 'status-success';
      statusText                                              =
TAPi18n.__('viewApiStatus_statusMessage_Success');
      statusIcon = 'mdi-checkbox-marked-circle';
      break;
    // Redirection code
    case 3:
      className = 'status-success';
```

```
      statusIcon = 'arrow-right-drop-circle';
      statusText = `

${TAPi18n.__('viewApiStatus_statusMessage_ErrorCodeText')}
          ${serverStatusCode}.

${TAPi18n.__('viewApiStatus_statusMessage_RedirectError')}
          `;
      break;
    // Client Error code
    case 4:
      className = 'status-warning';
      statusIcon = 'mdi-alert-circle';
      statusText = `

${TAPi18n.__('viewApiStatus_statusMessage_ErrorCodeText')}
          ${serverStatusCode}.

${TAPi18n.__('viewApiStatus_statusMessage_ClientError')}
          `;
      break;
    // Server Error code
    case 5:
      className = 'status-danger';
      statusIcon: 'mdi-close-circle';
      statusText = `

${TAPi18n.__('viewApiStatus_statusMessage_ErrorCodeText')}
          ${serverStatusCode}.

${TAPi18n.__('viewApiStatus_statusMessage_ServerError')}
          `;
      break;
    // The api monitoring is disable
    default:
      className = 'invisible';
      break;
  }

  return { className, statusText, statusIcon };
}
```

Appendix 8

Content of api.js

```
// Request /rest/v1/apis/:id/documents/
CatalogV1.addRoute('apis/:id/documents', {
  //  Remove  documentation  from  given  API  (:id)  either
completely or partially
  delete: {
    authRequired: true,
    swagger: {
      tags: [
        CatalogV1.swagger.tags.api,
      ],
      summary:  'Delete  identified  documentation  or  all
documentation.',
      description: descriptionApis.deleteDocumentation,
      parameters: [
        CatalogV1.swagger.params.apiId,
        CatalogV1.swagger.params.url,
```

```
      ],
      responses: {
        200: {
          description:    'API    documentation    updated
successfully',
          schema: {
            type: 'object',
            properties: {
              status: {
                type: 'string',
                example: 'Success',
              },
              data: {
                $ref: '#/definitions/apiResponse',
              },
            },
          },
        },
        400: {
          description: 'Bad Request. Erroneous or missing
parameter.',
        },
        401: {
          description: 'Authentication is required',
        },
        403: {
          description: 'User does not have permission',
        },
        404: {
          description: 'API is not found',
        },
      },
      security: [
        {
          userSecurityToken: [],
          userId: [],
        },
      ],
    },
    action () {
      // Get ID of API
      const apiId = this.urlParams.id;
      // Get User ID
      const userId = this.userId;
      // Get API document
      const api = Apis.findOne(apiId);

      // API must exist
      if (!api) {
        // API doesn't exist
        return errorMessagePayload(404, 'API with specified
ID is not found.');
      }

      // User must be able to manage API
      if (!api.currentUserCanManage(userId)) {
        return errorMessagePayload(403, 'User does not have
permission to this API.');
      }

      // Check if documentation exists
      const apiDoc = ApiDocs.findOne({ apiId });
```

```
      if (!apiDoc) {
        return errorMessagePayload(404, 'No documentation
exists for this API.');
      }

      // Check if link for openAPI documentation was given
      const documentUrl = this.queryParams.url;

      // Remove identified documentation link
      if (documentUrl) {
        // Check if it is openAPI documentation or external
documentation link
        if (documentUrl === apiDoc.remoteFileUrl) {
          // Matching link found as openAPI documentatin, try
to remove
          const removeResult = ApiDocs.update(
            { apiId },
            { $unset: {
              remoteFileUrl: '',
            },
            },
          );
          // If removal of openAPI document link failed
          if (removeResult === 0) {
            const message = 'OpenAPI Documentation link
removal failure.';
            return errorMessagePayload(500, message, 'url',
documentUrl);
          }
        } else if (apiDoc.otherUrl.includes(documentUrl)) {
          // Matching link found as external documentatin,
try to remove
          const removeResult = ApiDocs.update(
            { apiId },
            { $pull: { otherUrl: documentUrl } },
          );
          // If removal of external document link failed
          if (removeResult === 0) {
            const message = 'External Documentation link
removal failure.';
            return errorMessagePayload(500, message, 'url',
documentUrl);
          }
        } else {
          // No matching link found
          const message = 'Documentation link match not
found.';
          return errorMessagePayload(404, message, 'url',
documentUrl);
        }
      } else {
        // Remove all documentation, because no link
identified
        Meteor.call('removeApiDoc', apiId);
      }

      // Prepare data to response, extend it with
Documentation URLs
      const responseData = Object.assign(
        // API has not changed, use already fetched value
        api,
```

```
        // Get updated values of Documentation urls
        {
          externalDocumentation: api.otherUrl(),
          documentationUrl: api.documentationUrl(),
        });

      // OK response with API data
      return {
        statusCode: 200,
        body: {
          status: 'success',
          data: responseData,
        },
      };
    },
  },
});
```

Appendix 9

Content of catalog.js

```
apiMonitoring: {
      type: 'object',
      properties: {
        _id: {
          type: 'string',
          example: 'api-id-value',
        },
        enabled: {
          type: 'boolean',
          example: 'true',
        },
        apiMonitoringResultList: {
          type: 'array',
          items: {
            type: 'object',
            properties: {
              responses: {
                type: 'object',
                description: 'text',
                properties: {
                  date: {
                    type: 'string',
                    example: 'text',
                  },
                  server_status_code: {
                    type: 'string',
                    example: 'text',
                  },
                  url: {
                    type: 'string',
                    example: 'url',
                  },
                },
              },
            },
          },
        },
      },
    },
```