



Omar Samir Mohammad

Web Application for Generating Italian Fiscal Code

Codice Fiscale

Metropolia University of Applied Sciences

Bachelor of Engineering

Information and Communication Technology

29.12.2024

Abstract

Author(s): Omar Samir Mohammad
Title: Application for generating Italian Fiscal Code
Number of Pages: 24 pages
Date: 29 December 2024

Degree: Bachelor of Engineering
Degree Programme: Information and Communication Technology
Instructor(s): Janne Salonen, Head of IT Department

The purpose of this work is to serve as a development guide for a new developer who wants to learn how the Italian fiscal code works, its history, and how to create an Italian fiscal code using Spring Boot and Angular technologies. The work introduces the steps and algorithmic logic of creating the system, and the source code is available via a GitHub link to support learning.

Keywords: Web development, backend development, java, spring boot, Codice Fiscale, Spring Boot, Angular, Algorithms, Data Security, Identity Management.

The originality of this thesis has been checked using Turnitin Originality Check service.

Contents

1	Introduction	4
1.1	Why using Angular and Spring Boot?	4
2	Fiscal Code	6
2.1	What is the Italian Fiscal Code?	6
2.2	Background	6
2.3	Usage	7
2.4	Fiscal Code Structure	7
2.5	Comparison between the fiscal code and henkilötunnus	13
3	Technologies for the web application	13
3.1	Spring Boot	13
3.2	Angular	14
3.3	Reason behind choosing these technologies	14
3.4	Database	15
4	Web Application: Fiscal Code	16
4.1	The design	16
4.2	Functional Workflow	17
4.3	Technical Details and Implementation	18
5	Conclusions	20
5.1	What Was Achieved	21
5.2	Challenges and Lessons Learned	21
5.3	Ideas for Improvement	22
	References	23

1 Introduction

The Italian fiscal code or what is called codice fiscale in Italian language. The Italian fiscal code represents the Italian taxpayers' identification number. The Italian fiscal code is a unique taxpayer code that identifies all individuals in their dealings with the Italian authorities and is a mandatory requirement for a whole series of activities.

Every Italian gets a fiscal code at birth, while foreigners, who arrives in Italy, must apply to the Italian Tax Agency, or what is named in Italian Agenzia delle Entrate, to get their own fiscal code. (Taxing.it, 2023)

This thesis is aimed at freshmen software developers, specifically recent IT graduates, who are eager to learn the principles of full-stack web application development. It serves as a practical example to demonstrate coding with Angular and Spring Boot while providing an in-depth look at creating the Italian fiscal code system.

Taking the Italian fiscal code as a thesis since there are growing cooperation between Finnish companies in Italy. Since the Italian fiscal code is so important in each step of life in Italy starting on opening a bank account and signing an employment contract. Also the importance of detecting fake fiscal codes since there are growing scams in Italy and even in Finland nowadays. I see that any Finnish company should have at least the knowledge of how fiscal code is being calculated and generated.

1.1 Why using Angular and Spring Boot?

Using Angular and Spring Boot to create a web application for generating the Italian fiscal code is a practical choice for several reasons:

1. Full-Stack Development Experience:

Combining Angular and Spring Boot allows developers to build both the frontend and backend of the application. This approach provides a comprehensive understanding of full-stack development, which is valuable for creating complete web applications.

2. Angular's Dynamic User Interface

Angular is a development platform, that includes a component-based framework for building scalable web applications (Angular.io, 2023). It enables the creation of responsive user interfaces where users can input their personal information to generate the fiscal code. Angular's component-based architecture and two-way data binding make it suitable for developing interactive forms and real-time validations.

3. Spring Boot's Robust Backend Support

Spring Boot makes it easy to create stand-alone, production-grade Spring based Applications that you can "just run" (Spring.io, 2023a). Spring Boot is a powerful framework for building RESTful APIs in Java. It simplifies the development of backend services by providing features like auto-configuration and embedded servers. In this project, Spring Boot handles the backend logic, including implementing the fiscal code generation algorithm and managing client requests.

4. Scalability and Maintainability

Both Angular and Spring Boot promote modular development. Angular's component-based structure and Spring Boot's emphasis on convention over configuration lead to codebases that are easier to maintain and scale. This modularity is beneficial for projects that may grow in complexity over time.

In summary, using Angular and Spring Boot to develop a web application to generate fiscal code offers a balanced approach to learning full-stack development, leveraging the strengths of both frameworks to create a functional and educational application.

I also would like to highlight an important point of using REST. “REST embraces the precepts of the web, including its architecture, benefits, and everything else. This is no surprise, given that its author (Roy Fielding) was involved in probably a dozen specs which govern how the web operates. Since it has benefits like its suitability in actions of (GET, POST, PUT, DELETE, and others), caching, redirection and forwarding, and security (encryption and authentication)”. (Spring.io, 2023b)

2 Fiscal Code

2.1 What is the Italian Fiscal Code?

The Italian Fiscal Code is an essential tool serves to identify individuals residing in or interacting with Italy, irrespective of citizenship or residency status. It's more like the Finnish social security number, or what's known as Henkilötunnus in Finnish language. It was first designed for the Italian Tax Office. The fiscal code in Italy plays now an important role and serves multiple purposes, like identifying individuals in the health system or natural persons who act as parties in private contracts. Given its widespread use, it has become an essential aspect of life for both citizens and foreign residents alike.

2.2 Background

The tax identification number, fiscal code, provides a means of identification of foreign citizens in their relations with public authorities and other administrations. For individuals, it is determined on the basis of personal data and is made up of an expression of 16 alphanumeric characters. (Agenzia delle Entrate, 2023)

2.3 Usage

The fiscal code first issued by the Italian Revenue Agency (Agenzia delle Entrate).

After that, the Health Insurance Card (Tessera Sanitaria) also started to use the fiscal code and to identify all citizens entitled to services from the National Health Service (Servizio Sanitario Nazionale - SSN) as on (Agenzia delle Entrate, 2023).

2.4 Fiscal Code Structure

Let's find out how does each digit gets its value, according to (Agenzia delle Entrate, 2023). The tax code number for individuals is composed of a 16-character alphanumeric expression. The first fifteen characters are indicative of the personal data of each individual in the following order:

- Three alphabetic characters for the surname.
- Three alphabetic characters for the given name.
- Two numeric characters for the year of birth.
- One alphabetic character for the month of birth.
- Two numeric characters for the day of birth and gender.
- Four characters, including one alphabetic and three numeric, for the Italian municipality or foreign country of birth.

The sixteenth character, alphabetic, serves as a check digit.

Characters Indicative of the Surname:

Surnames consisting of multiple parts or otherwise separated or interrupted are treated as if they were written in a single uninterrupted sequence of characters.

For married women, only the maiden surname is considered.

- If the surname contains three or more consonants, the three characters taken are, in order, the first, second, and third consonant.
- If the surname contains two consonants, the three characters taken are, in order, the first and second consonant and the first vowel.
- If the surname contains one consonant and two vowels, the three characters taken are, in order, the consonant and the first and second vowels.
- If the surname contains one consonant and one vowel, the consonant and vowel are taken, in order, and the third character is assigned as the letter "X."
- If the surname consists of two vowels only, they are taken in order, and the third character is assigned as the letter "X."

Characters Indicative of the First Name:

Double, multiple, or compound given names are treated as if written in full, in an uninterrupted sequence of characters.

- If the first name contains four or more consonants, the three characters taken are, in order, the first, third, and fourth consonant.
- If the first name contains three consonants, the three characters taken are, in order, the first, second, and third consonant.

- If the first name contains two consonants, the three characters taken are, in order, the first and second consonants and the first vowel.
- If the first name contains one consonant and two vowels, the three characters taken are, in order, the consonant and the first and second vowels.
- If the first name contains one consonant and one vowel, the consonant and vowel are taken, in order, and the third character is assigned as the letter “X.”
- If the first name consists of two vowels only, they are taken in order, and the third character is assigned as the letter “X.”

Date, Gender, and Place of Birth:

The two numeric characters indicative of the year of birth are, in order, the tens and unit digits of the year. The alphabetic character corresponding to the month of birth is assigned according to the following table:

January = A	May = E	September = P
February = B	June = H	October = R
March = C	July = L	November = S
April = D	August = M	December = T

Table A – Conversion of the Month of Birth

The two numeric characters indicative of the day of birth and gender are determined as follows:

- For males, the day of birth is unchanged, ranging from one to thirty-one, with days one to nine preceded by zero.

- For females, the day of birth is increased by forty, resulting in numbers ranging from forty-one to seventy-one.

The four alphanumeric characters indicative of the Italian municipality or foreign country of birth consist of one alphabetic character followed by three numeric characters, according to the encoding established by the Agenzia del Territorio.

Individuals with Identical Alphanumeric Expressions (Homocodes)

When two or more individuals have personal data that generate the same tax code (homocodes), their codes are differentiated. This is achieved by systematically substituting one or more digits from the right among the seven numeric characters in the code with corresponding alphabetic characters according to the following table:

0 = L	3 = P	7 = T
1 = M	4 = Q	8 = U
2 = N	5 = R	9 = V
	6 = S	

Table B – Conversion of Digits into Alphabetic Characters (for Homocodes)

Check Character:

The sixteenth character acts as a check for the correct transcription of the first fifteen characters and is determined as follows:

1. Each of the fifteen characters is converted to a numeric value based on its position (even or odd) and the correspondence tables below.

A or 0 = 0	F or 5 = 5	K = 10	P = 15	U = 20
B or 1 = 1	G or 6 = 6	L = 11	Q = 16	V = 21
C or 2 = 2	H or 7 = 7	M = 12	R = 17	W = 22
D or 3 = 3	I or 8 = 8	N = 13	S = 18	X = 23
E or 4 = 4	J or 9 = 9	O = 14	T = 19	Y = 24
		Z = 25		

Table C – Conversion of Characters in Even Positions

A or 0 = 1	F or 5 = 13	K = 2	P = 3	U = 16
B or 1 = 0	G or 6 = 15	L = 4	Q = 6	V = 10
C or 2 = 5	H or 7 = 17	M = 18	R = 8	W = 22
D or 3 = 7	I or 8 = 19	N = 20	S = 12	X = 25
E or 4 = 9	J or 9 = 21	O = 11	T = 14	Y = 24
		Z = 23		

Table D – Conversion of Characters in Odd Positions

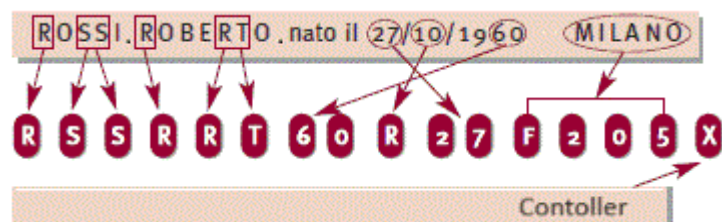
2. The numeric values are summed, and the total is divided by 26. The check character is obtained by converting the remainder of this division into the corresponding alphabetic character from the table below:

0 = A	5 = F	10 = K	15 = P	20 = U
1 = B	6 = G	11 = L	16 = Q	21 = V
2 = C	7 = H	12 = M	17 = R	22 = W
3 = D	8 = I	13 = N	18 = S	23 = X
4 = E	9 = J	14 = O	19 = T	24 = Y
		25 = Z		

Table E – Determination of the “Check-Digit”

Provisional Tax Code Number:

The tax administration may assign individuals a provisional tax code number. The provisional tax code number consists of an 11-digit numeric expression. The first seven digits represent the individual's registration number; digits eight through ten identify the office assigning the tax code; and the eleventh digit is the check digit.



Example: The above explains how an Italian Fiscal Code is created for Rossi Roberto, born in Milan on October 27, 1960, in Milan.

2.5 Comparison between the Italian fiscal code and the Finnish Henkilötunnus

the Italian fiscal code is more like the Finnish social security number, in the way how it's being used for each Italian resident to log in Italian public administrations and so on.

Of course, the fiscal code consists of 16 digits unlike the Finnish social security number that has only 11 digits. Whether an individual is a citizen residing in Italy, or an EU or non-EU citizen, the Italian fiscal code is essential. The fiscal code is a sequence of letters and numbers that uniquely identifies a person.

The code is issued by the Italian Revenue Agency (Agenzia delle Entrate); however Italian citizens obtain their fiscal code at birth, once registered in the Italian National Registry (Anagrafe Pubblica). Italian National Registry is the centralized database that simplifies registry services for people living in Italy.

3 Technologies for the web application

3.1 Spring boot:

Java Spring Boot (Spring Boot) is a tool that makes developing web applications and microservices with Java Spring Framework faster and easier. Java Spring Framework (Spring Framework) is a popular, open source, enterprise-level framework for creating stand-alone, production-grade applications that run on the Java virtual machine (JVM). (IBM, 2023)

According to (VMware Docs, 2023), Spring Boot is considered to be an excellent choice as a back-end technology to develop a web application that generate Italian fiscal code because of its advantages:

1. Spring Boot provides a radically faster and widely accessible getting-started experience for all Spring development.

2. Is opinionated out of the box while allowing flexibility as requirements diverge from the defaults.
3. Offers a range of non-functional features that are commonly needed across large classes of projects, such as embedded servers, security, metrics, health checks, and externalized configuration.
4. Avoids code generation (except when targeting native images) and eliminates the requirement for XML configuration.

3.2 Angular:

Angular is a web framework that empowers developers to build fast, reliable applications. Maintained by a dedicated team at Google, Angular provides a broad suite of tools, APIs, and libraries to simplify and streamline your development workflow. Angular gives you a solid platform on which to build fast, reliable applications that scale with both the size of your team and the size of your codebase. (Angular.dev. 2023)

Key aspects of choosing Angular as a platform (Angular.io. 2023):

1. A component-based framework for building scalable web applications.
2. A collection of well-integrated libraries that cover a wide variety of features, including routing, forms management, client-server communication, and more.
3. A suite of developer tools to help you develop, build, test, and update your code.

3.3 Reason behind choosing these technologies

This web application is designed as a Single Page Application (SPA), meaning that the client interacts with the entire functionality through a single, dynamic

webpage. As mentioned on (Spring.io, 2023c): “Spring Boot and Angular working together to provide a pleasant and secure user experience. It should be accessible to beginners with Spring and Angular, but there also is plenty of detail that will be of use to experts in either”, so we are going to make a web application using Spring Boot and Angular, to create the fiscal code generator. As many developers prefer microservices architecture to build their applications. Microservices is the right choice if a developer must build a scalable and enterprise-scale application, but if a developer just needs a small web application, then using microservices architecture will be an overkill as managing deployment and other aspects will be time consuming.

3.4 Database:

Under the General Data Protection Regulation (GDPR), obtaining explicit consent from individuals before collecting, processing, or storing their personal data is a fundamental requirement. Consent must be freely given, specific, informed, and unambiguous, signified by a clear affirmative action from the data subject. This is outlined in Article 4(11) of the GDPR, which defines consent as: **"Any freely given, specific, informed and unambiguous indication of the data subject's wishes by which he or she, by a statement or by a clear affirmative action, signifies agreement to the processing of personal data relating to him or her."** (GDPR-info.eu, 2023)

Additionally, Article 7 of the GDPR specifies conditions for consent, emphasizing that data controllers must be able to demonstrate that the data subject has consented to the processing of their personal data. It also states that consent should be presented in an intelligible and easily accessible form, using clear and plain language.

Implementing a web application that processes personal data, such as generating Italian fiscal codes, necessitates compliance with these GDPR requirements. Ensuring that no personal data is stored without explicit consent

aligns with the principles of data minimization and purpose limitation, thereby reducing the risk of data misuse and enhancing user privacy.

If it's necessary to store client data and their fiscal codes, any database management system could work, with the web application that we took here. However, instead of using a database management system for now, I was considering keeping the web app simple by using only two text files to collect the place of birth codes, which are part of the fiscal code. These files can be merged with the other Spring Boot files to keep the web app straightforward and easier to manage.

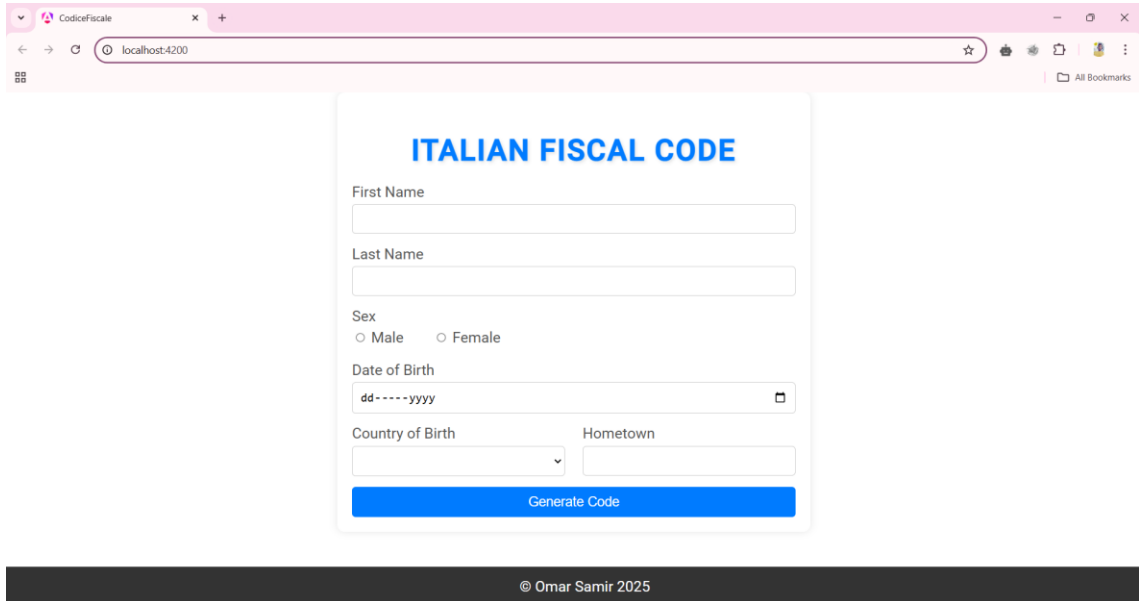
4 Web Application: Fiscal Code

4.1 The design

The Fiscal Code Generator Web Application was designed to provide a user-friendly interface for generating Italian fiscal codes. The design focuses on simplicity and usability, ensuring that users can efficiently input their personal information and retrieve their fiscal code.

The interface adheres to modern design principles:

1. **Minimalist Layout:** A clean, white background ensures clarity and reduces visual distractions.
2. **Form Accessibility:** Labels, input fields, and error messages are styled for easy readability. Fields are clearly marked for first name, last name, gender, date of birth, country of birth, and hometown.
3. **Responsive Design:** The web page is optimized for various screen sizes, ensuring usability on both desktop and mobile devices.



The image shows a web browser window with a single tab titled 'CodiceFiscale'. The address bar displays 'localhost:4200'. The main content area features a form with the heading 'ITALIAN FISCAL CODE' in blue. The form contains the following fields: 'First Name' (text input), 'Last Name' (text input), 'Sex' (radio buttons for 'Male' and 'Female'), 'Date of Birth' (text input with a date picker icon and placeholder 'dd-**-****'), 'Country of Birth' (dropdown menu), and 'Hometown' (text input). A prominent blue button labeled 'Generate Code' is positioned at the bottom of the form. Below the browser window, a black footer bar contains the text '© Omar Samir 2025'.

Figure 1. User-friendly interface for generating Italian fiscal codes.

4.2 Functional Workflow

Upon interacting with the application:

1. **User Input:** The user enters their details in the form, ensuring all mandatory fields are completed.
2. **Validation:** Angular's reactive forms provide real-time feedback for invalid or missing inputs.
3. **Submission:** When the "Generate Code" button is clicked, the form data is sent to the Spring Boot backend via an API call.
4. **Processing in Backend:** The backend validates the input data and processes it to generate the fiscal code based on Italian legal standards.
5. **Result Display:** The generated fiscal code is displayed on the same page, along with any relevant error messages (e.g., invalid hometown).

4.3 Technical Details and Implementation

The Fiscal Code Generator Web Application is built using a modular architecture to ensure scalability and maintainability. Below is a detailed breakdown of the technical implementation:

➤ **Frontend: Angular Framework**

The frontend is implemented in Angular, which provides a component-based architecture for building user interfaces. Key features of the frontend include:

1. Reactive Forms for Input Management:

- First Name and Last Name must have at least two characters and accept only alphabetic characters, spaces, and hyphens.
- Date of Birth must be a valid date within a realistic range.
- Gender is required, selectable via radio buttons.
- Error Feedback: Angular's reactive forms provide real-time feedback with error messages displayed beneath the input fields.

2. Dynamic Country Dropdown:

- The list of countries is fetched dynamically from an API or a local JSON file.
- The `HttpClient` service is used to retrieve the data, which is then sorted alphabetically and populated in the dropdown menu.

3. Event Handling

- The `onSubmit` function triggers an HTTP POST request to the backend with the user's input data.
- Real-time feedback is provided, displaying either the generated fiscal code or relevant error messages.

➤ **Backend: Spring Boot Application**

The backend is implemented in Spring Boot, which handles the business logic of generating the fiscal code. Key aspects include:

1. RESTful API

- A single endpoint `/api/person/generate-code` accepts POST requests containing the user's data in JSON format.
- The backend validates the input data and processes it according to the rules for Italian fiscal codes.

2. Error Handling

- Errors such as invalid hometowns or missing data are captured and returned as structured responses to the frontend.
- Example: If the hometown is invalid, the backend returns a custom error code, which is displayed to the user.

➤ **Integration: Country and Hometown Validation**

1. Country Data:

- The application uses an external API to validate the selected country of birth. There's a possibility to local JSON file as well.

- Used API: <https://restcountries.com/v3.1/all>.

2. Hometown Validation:

- A separate validation logic ensures the hometown exists in the database or an external service.
- If the hometown is invalid, an error message is returned.

➤ Styling

1. CSS and Bootstrap:

- **Bootstrap** is used to style the form and make it responsive.
- Additional CSS ensures a modern and visually appealing design, including a custom header and footer.

2. Responsive Design:

- The application adjusts seamlessly to various screen sizes, making it accessible on mobile devices, tablets, and desktops.

5 Conclusions

This project aimed to create a web application for generating Italian fiscal code using Angular and Spring Boot. The goal was to show how to build a complete web application while helping beginners understand the basics of full-stack development. The project focused on the importance of the fiscal code in Italy and how it is generated.

The application was designed to be simple and easy to use. Angular was used for the frontend to make a dynamic and user-friendly interface, while Spring Boot handled the backend, including the logic for creating fiscal codes. The

application was built as a Single Page Application (SPA) to provide a smooth user experience.

5.1 What Was Achieved

1. **Full-Stack Development:** The project successfully combined Angular and Spring Boot. Angular created an intuitive user interface, and Spring Boot processed the data to generate fiscal codes.
2. **Fiscal Code Generation:** The application followed Italian legal standards to calculate fiscal codes. It encoded details like name, date of birth, gender, and place of birth into the correct 16-character format.
3. **Data Privacy:** The project followed GDPR rules by not saving personal data. All data was processed temporarily to ensure user privacy.
4. **Validation and Errors:** Input validation on both the frontend and backend ensured reliable results. Clear error messages helped users fix any issues in their input.
5. **Educational Value:** This project provided a practical example of full-stack development, helping new developers learn key skills like building APIs and responsive interfaces.

5.2 Challenges and Lessons Learned

Some challenges included ensuring consistent data handling between the frontend and backend, understanding the rules for generating fiscal codes, and integrating external APIs for validating countries and hometowns. Solving these problems improved the application's reliability and taught valuable lessons about system integration.

5.3 Ideas for Improvement

1. Database Use: Adding a database would make the app more scalable and allow features like user accounts and history tracking.
2. Better User Interface: Features like autocomplete for hometowns and live feedback while filling out the form could improve usability.
3. Language Support: Adding Italian and Finnish language options would make the app more user-friendly for a wider audience.
4. Deployment: Hosting the app on a cloud platform would allow users to access it online and test it in real-world conditions.

References

1. Agenzia delle Entrate. 2023. Codice Fiscale e Tessera Sanitaria Services. Rome. Agenzia delle Entrate.

<https://www.agenziaentrate.gov.it/portale/web/guest/agenzia/amministrazione-trasparente/servizi-erogati/carta-servizi/i-nostri-servizi/area-identificazione-del-contribuente/codice-fiscale-e-tessera-sanitaria>

2. Agenzia delle Entrate. 2023. Informazioni Codificazione PF. Rome. Agenzia delle Entrate.

https://www.agenziaentrate.gov.it/portale/web/guest/schede/istanze/richesta-ts_cf/informazioni-codificazione-pf

2 no use since in Italian language

3. Agenzia delle Entrate. 2023. Tax Identification Number for Foreign Citizens. Rome. Agenzia delle Entrate.

<https://www.agenziaentrate.gov.it/portale/web/english/nse/individuals/tax-identification-number-for-foreign-citizens>

4. Angular.io. 2023. Angular Documentation Version 17. Mountain View. Angular Project.

<https://v17.angular.io/docs>

5. Angular.dev. 2023. Overview of Angular. Mountain View. Angular Project.

<https://angular.dev/overview>

6. Angular.io. 2023. What is Angular?. Mountain View. Angular Project.

<https://v17.angular.io/guide/what-is-angular>

7. GDPR-info.eu. 2023. Article 4 GDPR. Brussels. GDPR Info.

<https://gdpr-info.eu/art-4-gdpr/>

8. IBM. 2023. Java Spring Boot Overview. Armonk. IBM.

<https://www.ibm.com/topics/java-spring-boot>

9. Spring.io, 2023a. Spring Boot Project Documentation. Palo Alto. VMware.

<https://spring.io/projects/spring-boot>

10. Spring.io, 2023b. REST Tutorial by Spring. Palo Alto. VMware.

<https://spring.io/guides/tutorials/rest>

11. Spring.io, 2023c. Spring Security and Angular JS Tutorial. Palo Alto. VMware.

<https://spring.io/guides/tutorials/spring-security-and-angular-js>

12. Taxing.it. 2023. Italian Fiscal Code Number - Tax Code (TIN). Milan. Taxing.it.

<https://taxing.it/italian-fiscal-code-number-tax-code-tin/>

13. VMware Docs. 2023. Spring Boot Documentation. Palo Alto. VMware.

<https://docs.spring.io/spring-boot/index.html>