

Opc Server with Milo and Spring Boot

Documentazione di progetto per la materia Informatica Industriale

- [Introduzione](#)
- [Perché Spring](#)
- [Descrizione del progetto](#)
- [Funzionamento](#)
- [Annex 1: GitHub](#)
 - [Step1: Creare il progetto su github](#)
 - [Step2: Readme and .gitignore](#)
 - [Step3: Repo Download](#)
- [Annex 2: Camel Demo with Spring Boot](#)
 - [Step1: Spring Init](#)
 - [Step2: Importare il progetto Maven dentro Eclipse](#)
 - [Step3: Apache Milo](#)

Introduzione

Nell'ambito del corso di Informatica Industriale è stato richiesto di sviluppare un Server OpcUa.

La presente documentazione descrive come poter implementare un Server Opc utilizzando come standard SDK per implementare lo stack protocollare [Eclipse Milo](#).

L'idea di questo progetto è quella di implementare il server Opc utilizzando [Spring Boot](#), un framework di sviluppo molto diffuso anche in ambito java enterprise.

Perché Spring

//TODO

Descrizione del progetto

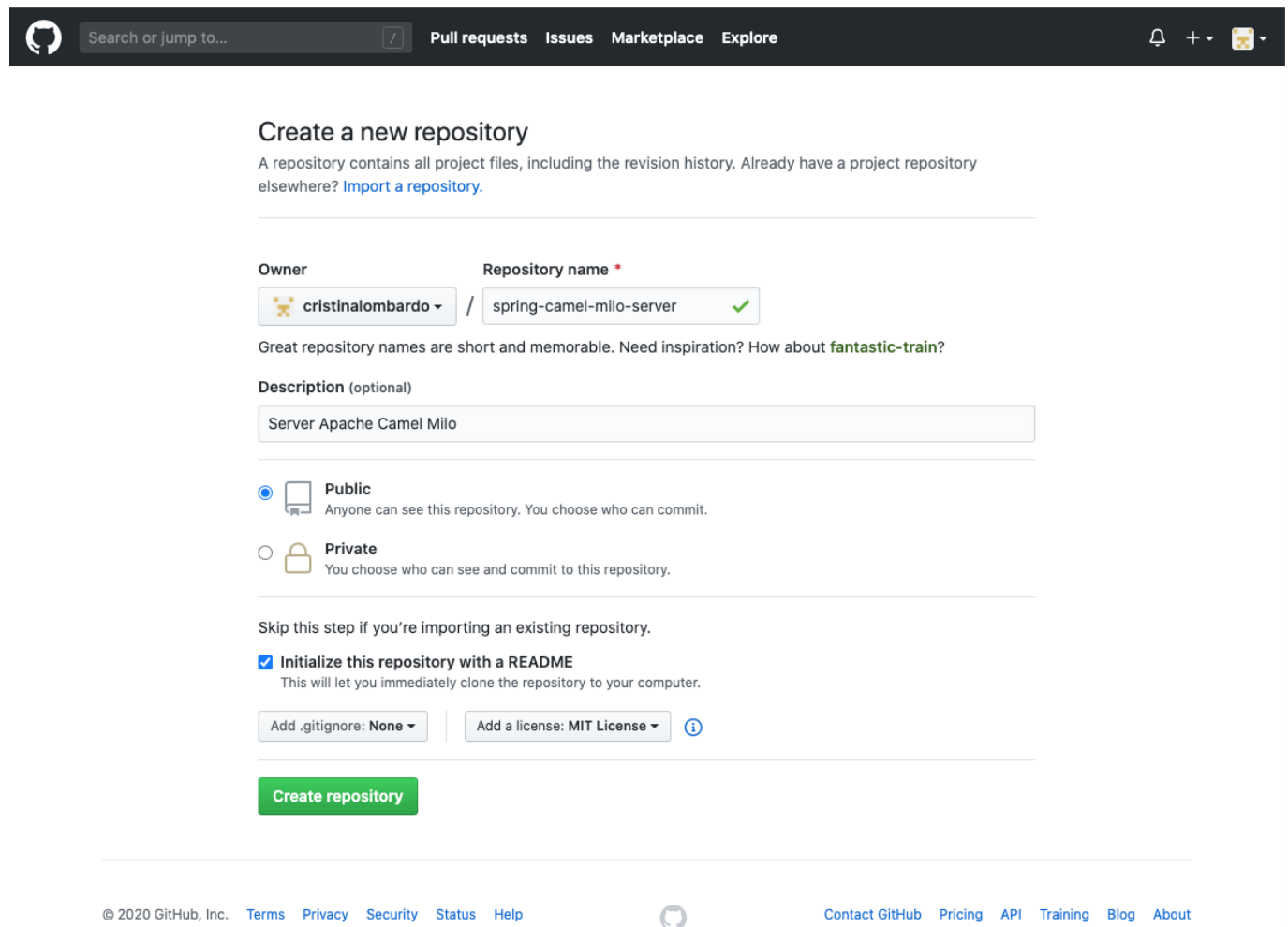
//TODO

Funzionamento

//TODO

Annex 1: GitHub



Step1: Creare il progetto su github



Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)


Owner **Repository name ***


 cristinalombardo / spring-camel-milo-server 

Great repository names are short and memorable. Need inspiration? How about **fantastic-train**?

Description (optional)


Server Apache Camel Milo

☒  **Public**
Anyone can see this repository. You choose who can commit.


☐  **Private**
You choose who can see and commit to this repository.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**
This will let you immediately clone the repository to your computer.

Add .gitignore: **None** | Add a license: **MIT License** 

Create repository

© 2020 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)  [Contact GitHub](#) [Pricing](#) [API](#) [Training](#) [Blog](#) [About](#)

Step2: Readme and .gitignore

Aggiungere Readme e .gitignore

Come template del .gitignore è stato scelto Maven.

Skip this step if you're importing an existing repository.

☒ **Initialize this repository with a README**

This will let you immediately clone the repository to your computer.

Add .gitignore: **None** ▼

Add a license: **MIT License** ▼



.gitignore

Mave

Maven

2020 GitHub, Inc. [Terms](#) [Privacy](#) [Security](#) [Status](#) [Help](#)



[Conte](#)

Step3: Repo Download

Creare il repository e clonarlo in locale con il programma preferito di gestione del git.

Annex 2: Camel Demo with Spring Boot

Prerequisiti:

- Java 9+
- Eclipse per j2ee con Spring Tools Suite Plugin installato

Step1: Spring Init

Per inizializzare il progetto utilizzeremo Spring Initializr

[Go to Spring initializr](#)

Inizializzare il progetto in accordo alla seguente immagine

The screenshot shows the Spring Initializr web interface. On the left, there is a sidebar with a hamburger menu icon and a Twitter icon. The main content area is divided into two columns. The left column contains the 'Project' and 'Language' sections. The 'Project' section has radio buttons for 'Maven Project' (selected) and 'Gradle Project'. The 'Language' section has radio buttons for 'Java' (selected), 'Kotlin', and 'Groovy'. Below this is the 'Spring Boot' section with radio buttons for versions: '2.3.1 (SNAPSHOT)', '2.3.0' (selected), '2.2.8 (SNAPSHOT)', '2.2.7', '2.1.15 (SNAPSHOT)', and '2.1.14'. The 'Project Metadata' section includes fields for 'Group' (com.github.cristinalombardo), 'Artifact' (opcserver), 'Name' (Spring Boot OPC Server), 'Description' (OPC Server using Apache Camel Milo into Spring Boot), and 'Package name' (com.github.cristinalombardo.opcserver). The 'Packaging' section has radio buttons for 'Jar' (selected) and 'War'. The 'Java' section has radio buttons for '14', '11' (selected), and '8'. The right column contains the 'Dependencies' section with a button 'ADD DEPENDENCIES... 8 + 8'. Below this is the 'Spring Boot DevTools' section, which is highlighted with a green 'DEVELOPER TOOLS' tag. It includes a description: 'Provides fast application restarts, LiveReload, and configurations for enhanced development experience.' At the bottom of the form, there are three buttons: 'GENERATE ⌘ + ↵', 'EXPLORE CTRL + SPACE', and 'SHARE...'. The top right corner of the form has a settings icon and a dark mode icon.

Per inizializzare il progetto abbiamo utilizzato le seguenti configurazioni:

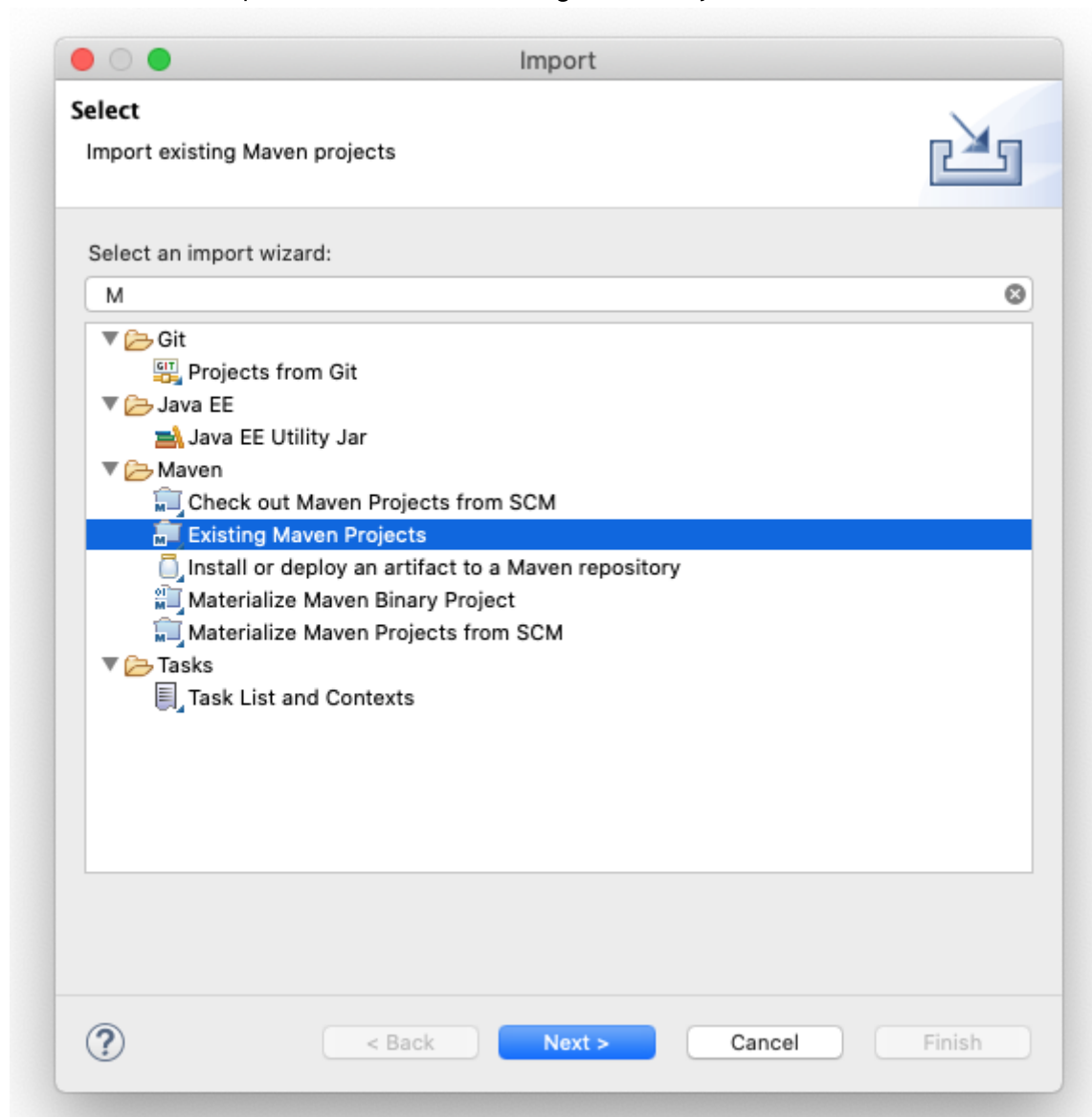
- Maven: Utilizzeremo Maven per compilare il progetto
- Spring Boot 2.3.0: Versione Stabile di Spring Boot al momento di questo tutorial
- Packaging Jar: Si vuole realizzare un'applicazione standalone
- Java 11: Apache Camel Milo richiede Java 9+
- Dependencies Spring Boot DevTools: dipendenza utile durante lo sviluppo

A questo punto è possibile scaricare il progetto cliccando sul tasto GENERATE.
Una volta scaricato il progetto è stato copiato nella root del repository e decompattato.

Step2: Importare il progetto Maven dentro Eclipse

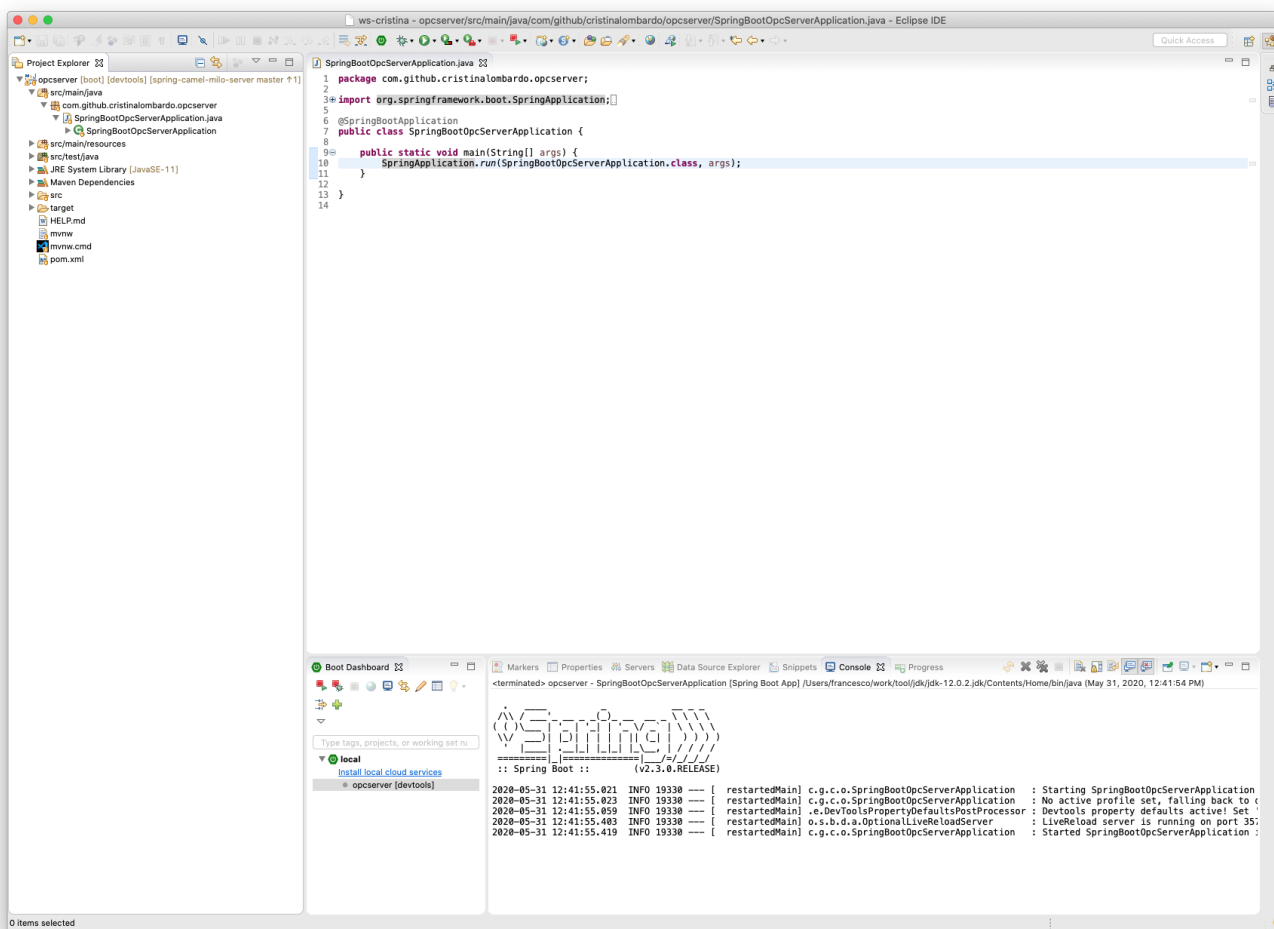
Per importare il progetto i passi sono:

1. Aprire Eclipse
2. Cliccare su File->Import... e selezionare Existing Maven Project



3. Selezionare la cartella dove risiede il progetto e cliccare su Finish

A questo punto il progetto dovrebbe presentarsi nel seguente modo:



Step3: Apache Milo

In questo step aggiungeremo le librerie Apache Camel Milo.

1. Apriamo il file `pom.xml`
2. Aggiungiamo la seguente dipendenza al file

```

<dependency>
  <groupId>org.apache.camel.springboot</groupId>
  <artifactId>camel-milo-starter</artifactId>
  <version>${camel.milo.version}</version>
  <!-- use the same version as your Camel core version -->
</dependency>

```

La versione di Camel viene inserita nella sezione `<properties>` del `pom.xml`

```

<camel.milo.version>3.0.0</camel.milo.version>

```