

Criando documentações automáticas para API Rest em Spring via Swagger

Configurando o pom.xml

coloque as seguintes dependencias:

```
<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger2</artifactId>
    <version>2.6.1</version>
</dependency>

<dependency>
    <groupId>io.springfox</groupId>
    <artifactId>springfox-swagger-ui</artifactId>
    <version>2.6.1</version>
</dependency>
```

Configurando o SwaggerConfig

crie um package chamado config em seu projeto, dentro dele crie uma classe chamada SwaggerConfig e configure segundo ao seguinte modelo.

```
import org.springframework.context.annotation.Bean;
import org.springframework.context.annotation.Configuration;
import springfox.documentation.builders.ApiInfoBuilder;
import springfox.documentation.builders.PathSelectors;
import springfox.documentation.builders.RequestHandlerSelectors;
import springfox.documentation.service.ApiInfo;
import springfox.documentation.service.Contact;
import springfox.documentation.spi.DocumentationType;
import springfox.documentation.spring.web.plugins.Docket;
import springfox.documentation.swagger2.annotations.EnableSwagger2;

@EnableSwagger2
@Configuration
public class SwaggerConfig {

    @Bean
    public Docket docket(){
        return new Docket(DocumentationType.SWAGGER_2)
```

```

        .select()
        .apis( RequestHandlerSelectors.basePackage
("generation.org.blogPessoal.controller") )
        .paths(PathSelectors.any())
        .build()
        .apiInfo(apiInfo());
    }

    private ApiInfo apiInfo(){
        return new ApiInfoBuilder()
            .title("Blog Pessoal")
            .description("API do Projeto de blog pessoal")
            .version("1.0")
            .contact(contact())
            .build();
    }

    private Contact contact(){
        return new Contact("Lucas Capelotto",
            "https://github.com/LucasCapSilva",
            "Instrutor full stack Angular Spring");
    }
}

```

No campo **.apis(**

RequestHandlerSelectors.basePackage("generation.org.blogPessoal.controller")) você deve definir o package do seu controller para que o swagger mapeie os controller para a documentação.

No campo **.title("Blog Pessoal")** você pode definir o título para a sua aplicação ser apresentada na documentação.

No campo **.description("API do Projeto de blog pessoal")** você pode definir uma descrição para a sua aplicação.

No campo **Contact("Lucas Capelotto",**

["https://github.com/LucasCapSilva",](https://github.com/LucasCapSilva)

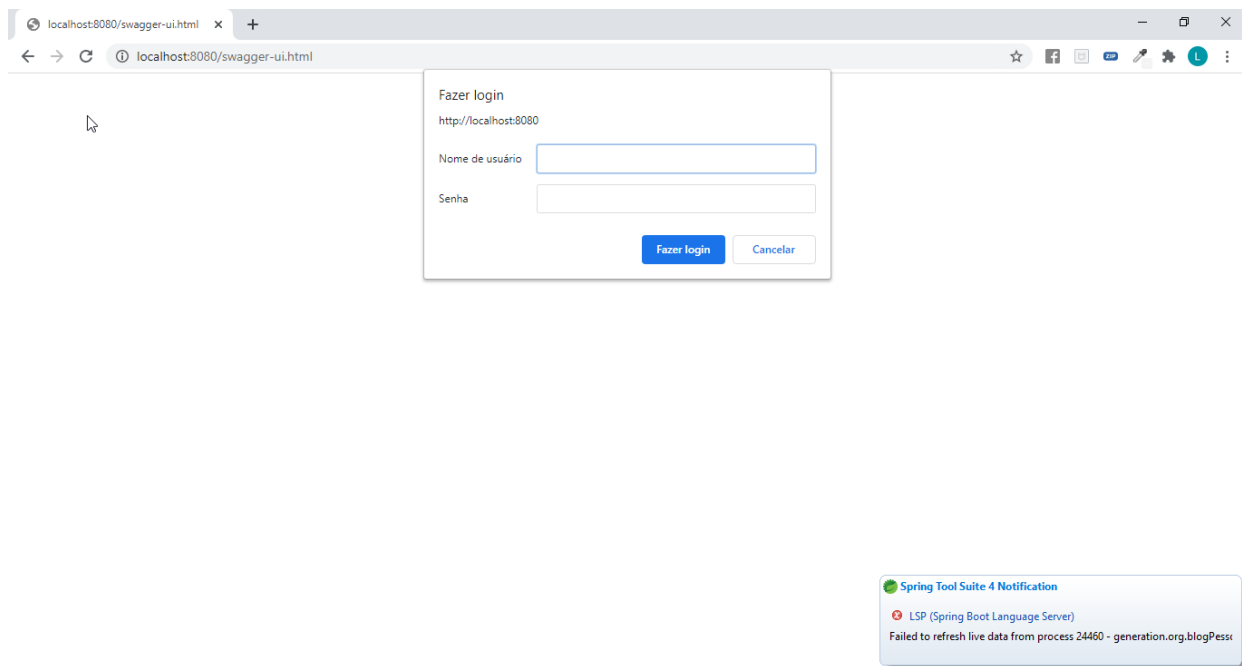
"Instrutor full stack Angular Spring"); você pode definir as informações de contato do desenvolvedor dessa aplicação.

salve todos arquivos e inicie a sua aplicação.

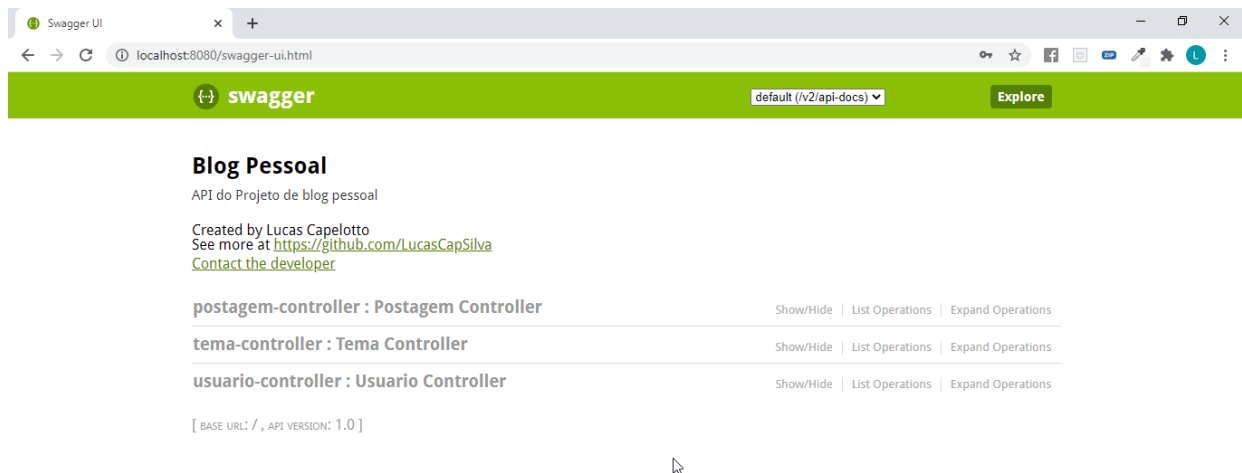
entre no seu navegador e digite o seguinte caminho:

<http://localhost:8080/swagger-ui.html> para abrir a sua documentação.

Em aplicações com camadas de segurança você precisara logar com um usuário existente antes de iniciar a sua documentação em swagger, coloque um usuario cadastrado para fazer o login.



E pronto sua documentação em Swagger esta funcionando.



Swagger UI

localhost:8080/swagger-ui.html#/postagem-controller/postUsingPOST

API do Projeto de blog pessoal

Created by Lucas Capelotto
See more at <https://github.com/LucasCapSilva>
[Contact the developer](#)

postagem-controller : Postagem Controller

Show/Hide | List Operations | Expand Operations

GET /postagem GetAll

POST /postagem post

Response Class (Status 200)
OK

Model | Example Value

```
{
  "data": "2020-09-04T10:29:49.118Z",
  "id": 0,
  "tema": {
    "descricao": "string",
    "id": 0,
    "postagem": [
      {}
    ]
  }
}
```

Response Content Type */* ▼

Parameters

Parameter	Value	Description	Parameter	Data Type
-----------	-------	-------------	-----------	-----------