



Como implementar na sua
arquitetura?

Yelken Gonzales

1. 13 anos de experiência com desenvolvimento JAVA e sistemas distribuídos
2. Participante ativo em Hackathons (+ de 37)
3. Formado em Sistemas de Informação e Análise e desenvolvimento de sistemas
4. Pós graduado em Segurança da Informação - UNIBRATEC
5. MBA em Arquitetura de Software (2019)
6. OCJA / OCJP

O que é segurança de software?

1. É um estado (Ou sua aplicação está segura ou não está)
2. Confidencialidade (Garantir o acesso daquela área ou informação apenas a pessoas autorizadas)
3. Integridade (A área do software ou dado não podem ser alteradas ou excluídas sem a devida autorização)
4. Disponibilidade (Acesso ao software ou ao dado apenas por pessoas autorizadas)

Requisitos para um software seguro

1. Senha e outros campos de entrada de dados sensíveis necessitam ser mascarados.
2. Transport Layer Security (TLS) como Secure Socket Layer (SSL) deve ser colocado em prática para proteger contra ameaças internas de Man in the Middle (MITM)
3. O uso de protocolos reconhecidamente inseguros como, por exemplo, File Transfer Protocol (FTP) para transmitir credenciais de contas em texto claro.
4. Arquivos de log não devem armazenar qualquer informação sensível como definido pelo negócio.

Requisitos para um software seguro

1. Todos os formulários de entrada e query strings necessitam ser validadas frente a um conjunto de entradas aceitáveis, antes do software aceitá-los para processamento.
2. O software a ser publicado deve ser disponibilizado juntamente com o checksum e a função hash usada para computar o checksum.
3. Todos os personagens não humanos como um sistema ou processos batch devem ser identificados, monitorados e impedidos de alteração de dados.

Spring Security

Veio para juntar várias soluções que o JAVA já oferecia em apenas um FRAMEWORK:

Authorization Service (JAAS) ou Java EE Security

Criado em 2003 como Acegi Security

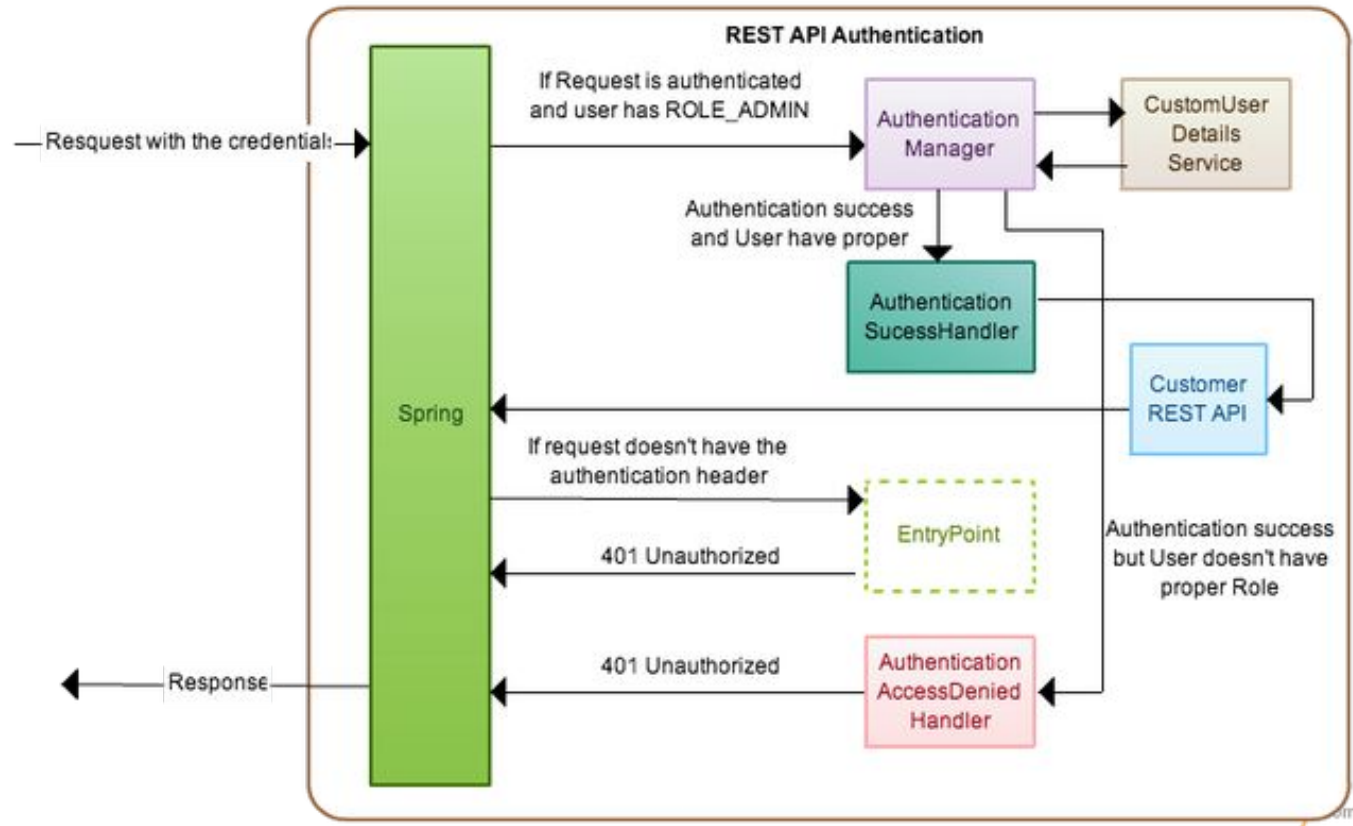
Lançado oficialmente em 2008 como Spring Security

Versão 5.2.0 (STABLE)

Funcionalidades principais

1. Fornece autenticação e autorização para sistemas corporativos
2. Segmentar os usuários do sistema em classes de usuário;
3. Atribuir níveis de autorização à papéis de usuário;
4. Atribuir papéis de usuário à classes de usuário;
5. Aplicar regras de autenticação globais para recursos de aplicação;
6. Aplicar regras de autorização à todos os níveis da arquitetura da aplicação;
7. Prevenir tipos de ataques comuns que manipulam ou roubam sessões do usuário.

Arquitetura do Spring Security com REST API



Falando em arquitetura
Algumas dicas!

Como implementar o Spring Security?

Recomendo (<https://start.spring.io/>)

Supondo que você já rode com Spring Boot

Opa! O que é Spring Boot

Facilita o processo de configuração e publicação das nossas aplicações

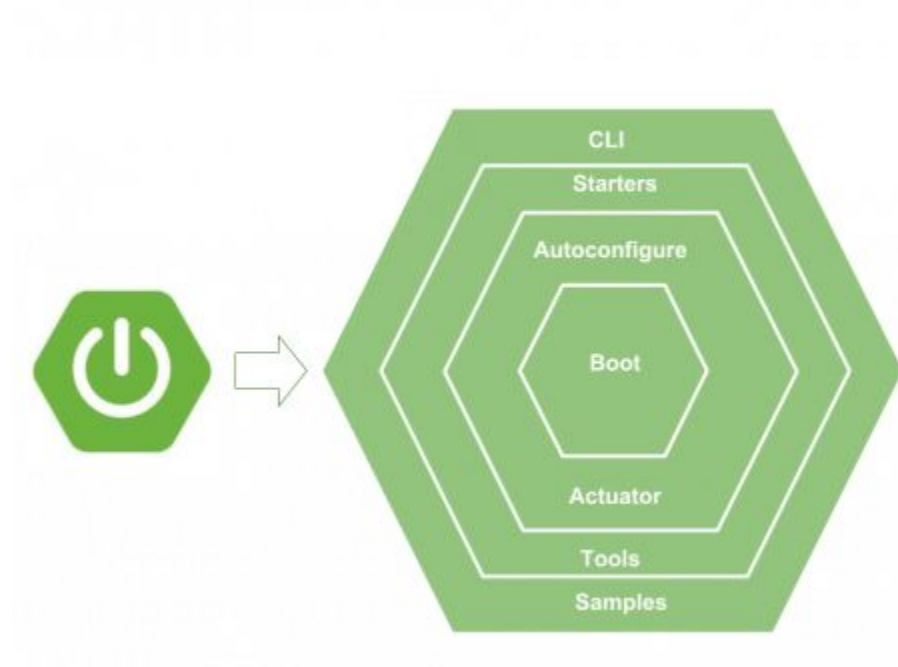
"O Spring Boot, de forma diferente do Roo (acima de tudo não há geração de código), busca solucionar a complexidade da inicialização e gerenciamento de dependências de um projeto com Spring, além de tratar de maneira coesa e eficiente a questão da configuração, fazendo uso extensivo de *Convention Over Configuration*."

"O Spring Boot pode ser considerado um plugin para a ferramenta de building, seja ela o Maven ou o Gradle. Seus principais objetivos são gerenciar dependências de maneira opinativa e automática, e simplificar a execução do projeto em tempo de desenvolvimento e depuração."

Spring Boot

```
mvn spring-boot:run
```

Arquitetura do Spring Boot



Spring Security (Dependência)

```
<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-oauth2-client</artifactId>
</dependency>
<dependency>
  <groupId>org.springframework.security</groupId>
  <artifactId>spring-security-oauth2-jose</artifactId>
</dependency>
```


Spring Security (application.properties)

```
spring.security.oauth2.client.registration.google.client-id=<your client id>  
spring.security.oauth2.client.registration.google.client-secret=<your client secret>  
  
spring.security.oauth2.client.registration.facebook.client-id=<your client id>  
spring.security.oauth2.client.registration.facebook.client-secret=<your client secret>
```

Spring Security

URLS geradas

`http://localhost:8081/login/oauth2/client/google`

`http://localhost:8081/login/oauth2/client/facebook`

Spring Security

Configuração da classe

```
@Configuration
```

```
public class SecurityConfig extends WebSecurityConfigurerAdapter {
```

```
    @Override
```

```
    protected void configure(HttpSecurity http) throws Exception {
```

```
        http.authorizeRequests()
```

```
            .anyRequest().authenticated()
```

```
            .and()
```

```
            .oauth2Login();
```

```
    }
```

```
}
```

Spring Security Feito!

Login with OAuth 2.0

[Facebook](#)

[Google](#)

Spring Security

Login with OAuth 2.0

[Facebook](#)

[Google](#)

ATIVEM O LINKEDIN

OBRIGADO GENTE
[linkedin.com/yelken](https://www.linkedin.com/company/yelken)