

## Instructions

Add Spring Security to your service (you can use any service that you have up and running)

Remember that these are the steps that you need to follow:

### Part 1

1) Add the following dependency to your project:

```
<dependency>
  <groupId>org.springframework.boot</groupId>
  <artifactId>spring-boot-starter-security</artifactId>
</dependency>
```

2) Add the following to your application.yml (server side)

```
spring:
  security:
    user:
      name: app-user
      password: c1e3d0fa-5aa8-4be7-83f7-32974dd617bf
```

(obviously, username and password can be set to anything you like)

Test your service and observe that the app now requires a username and password.

### Part 2

Make adjustments to your client service (client service calling your service) so that it can call the now secure service

1) Add the following code to your client service

```
@Value("${server-service.service-name}")
private String serverServiceName;

@Value("${server-service.username}")
private String username;

@Value("${geography-service.password}")
private String password;

private List<CountryResponse> countryListCache = new ArrayList<>();

@Override
@HystrixCommand(fallbackMethod = "defaultGetAllCountries")
public List<CountryResponse> getAllCountries() {
    String url = lookupUrlFor(serverServiceName) + "/countries";
    countryListCache = Arrays.asList(restTemplate.exchange(url, HttpMethod.GET,
        createHttpEntity(), CountryResponse[].class).getBody());
}
```

```

        return countryListCache;
    }

    public List<CountryResponse> defaultGetAllCountries() {
        return countryListCache;
    }

    private HttpEntity<Object> createHttpEntity() {
        HttpHeaders headers = new HttpHeaders();
        headers.setAccept(Arrays.asList(MediaType.APPLICATION_JSON));
        headers.setBasicAuth(username, password);

        return new HttpEntity<>(headers);
    }

    private String lookupUrlFor(String appName) {
        InstanceInfo instanceInfo = eurekaClient.getNextServerFromEureka(appName, false);
        return instanceInfo.getHomePageUrl();
    }
}

```

2) Add the following configuration to your application.yml

```

server-service:
  service-name: simple-geography-service
  username: app-user
  password: c1e3d0fa-5aa8-4be7-83f7-32974dd617bf

```

Hint: Pay attention that your username and password should match the ones entered in part 1.

Now, test the client app and observe that everything is working again.