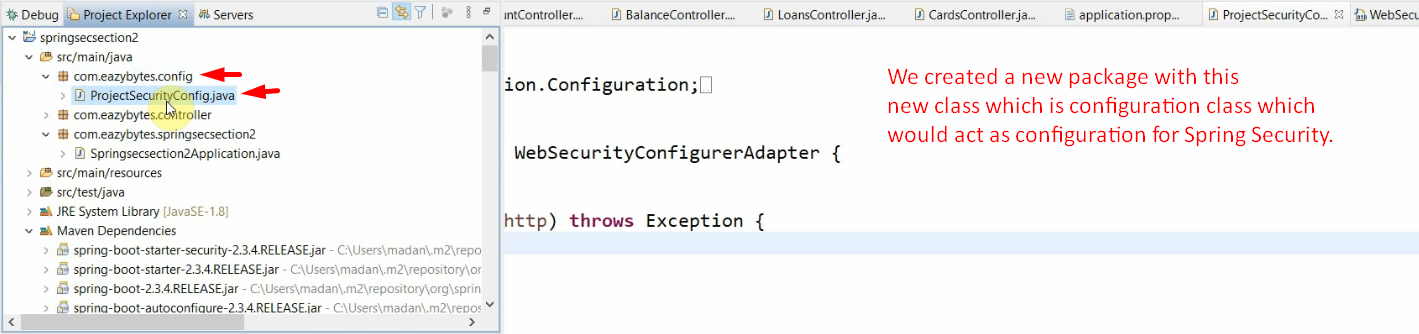
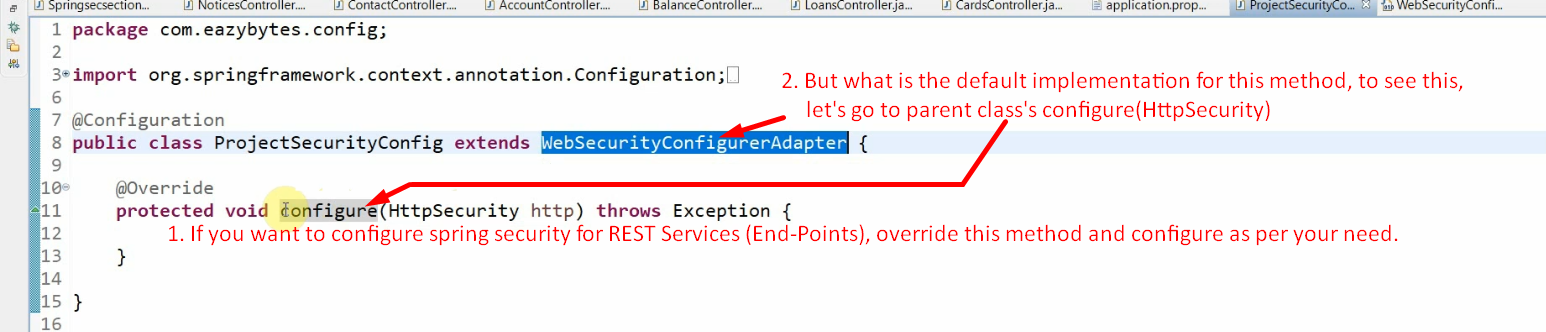
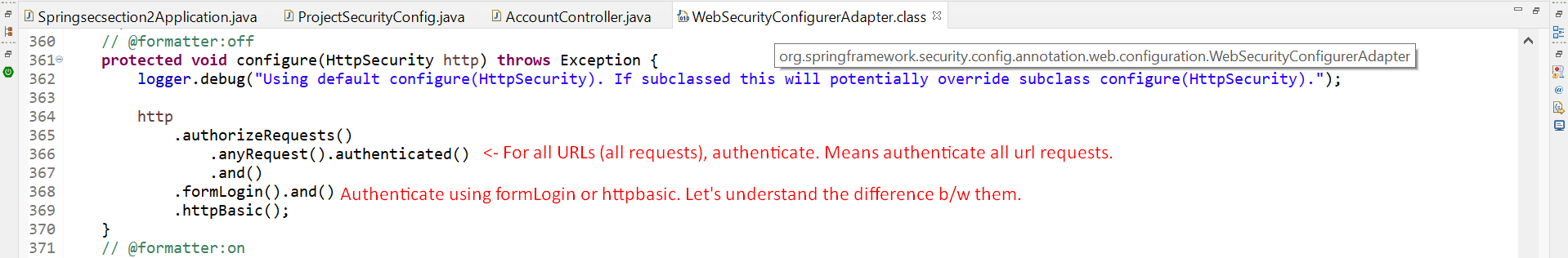
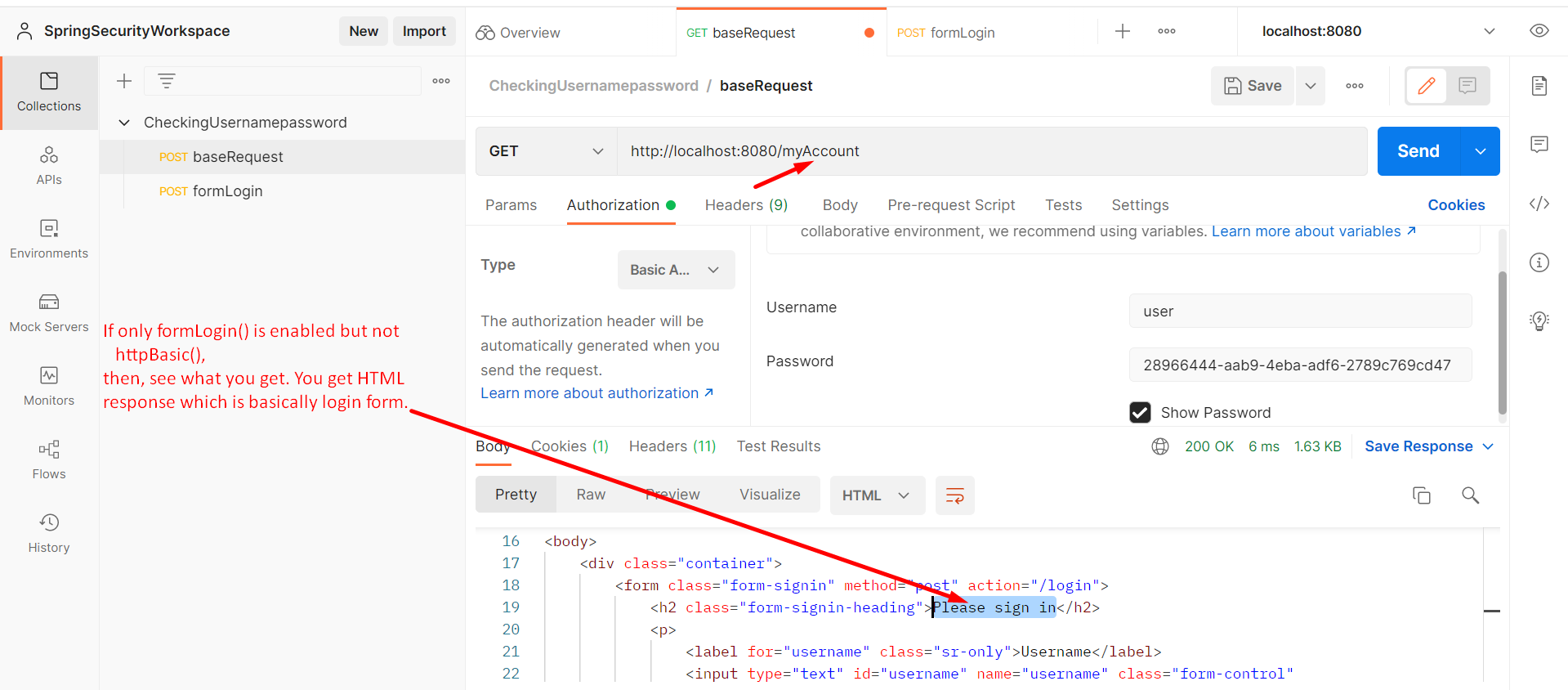
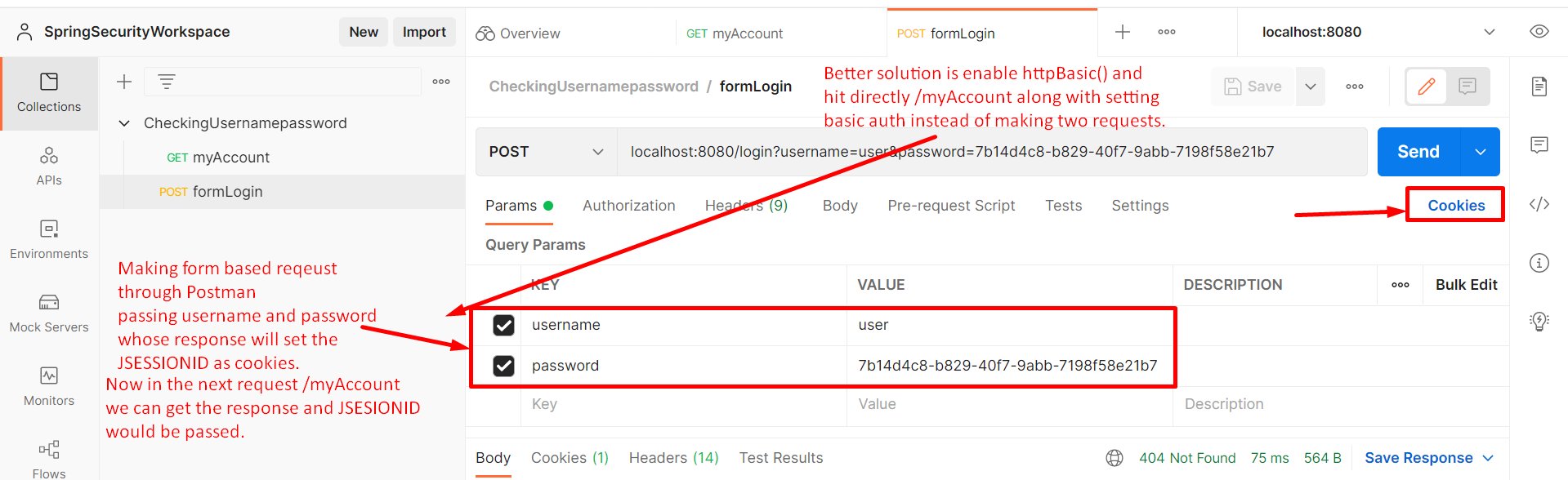
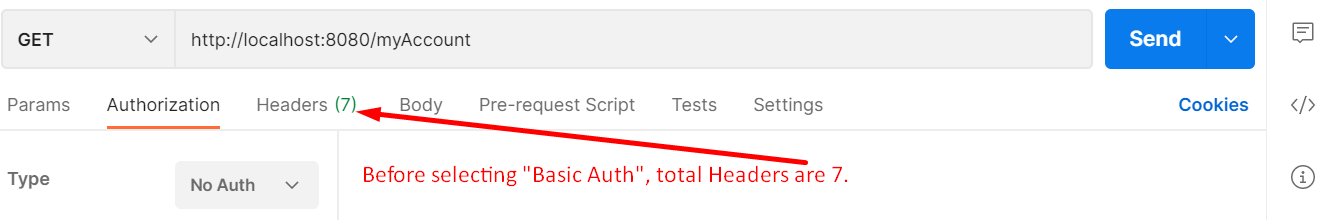
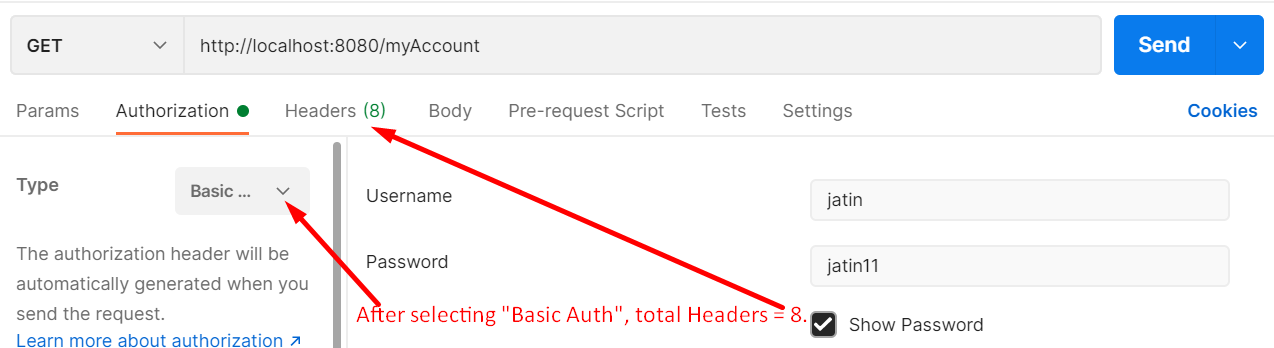
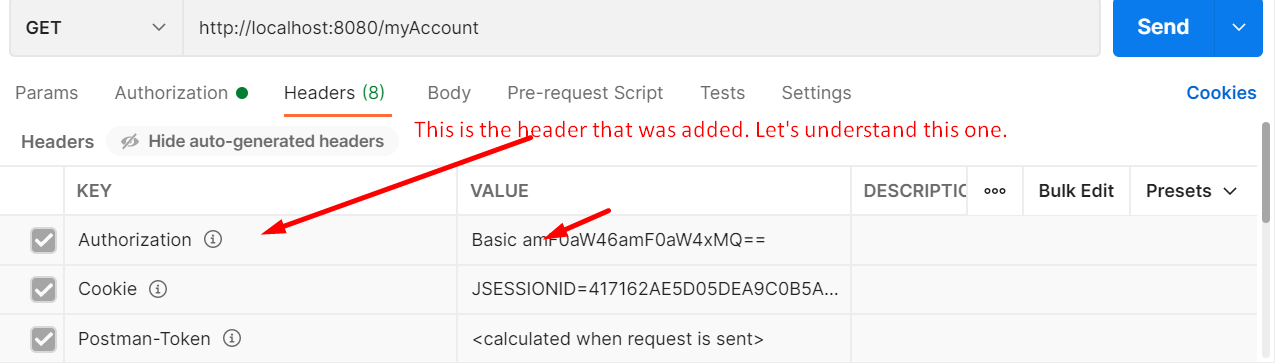
1. **Agenda**: By default, Spring Security secures all the end-points. Let’s prove it.
2. 
3. **@Configuration**:
   1. Indication to Spring Boot to consider this class as configuration class while booting up the app.
4. **WebSecurityConfigurerAdapter**:
   1. Extend and override its methods as per your need when you’re going to customize spring security configurations.
5. 
6. 
   1. See, we can send request either from browser or not from a browser such as from postman, RestTemplate, some other programming language.
   2. formLogin():
      1. User is making request from browser.
      2. Still user is not verified and making request for some URL.
      3. Spring Security identifies that the user is still not authenticated and user is browser user.
      4. Spring Security sees the HTTP configuration too. It finds one of the two cases.
         1. formLogin() **is configured**:
            1. If formLogin() is configured, a login page would be returned.
            2. No matter if httpBasic is also configured because the user is browser user.
         2. formLogin() **is not configured but httpBasic is configured**:
            1. HttpBasic Popup would be displayed for user to enter credentials.
   3. httpBasic():  
         
      But there is no use for the above login form response as we can’t use that form in Postman.   
      But only loginForm() is enabled, it doesn’t mean, through Postman we can’t login. See the following slide.  
        
      1. Take an example of POSTMAN where we can’t enter login credentials via html page.
      2. So for such kinds of situations, we need to enable http basic authentication.
      3. 
      4. 
      5. 
      6. How credentials are passed in case of HTTP Basic.
         1. First this format (username:password ) is encoded using Base64 🡺 **Base64(username:password)**
         2. Then, a header is added named “Authorization” with the above as its value.

|  |  |
| --- | --- |
| **Header** | **Value** |
| Authorization | Basic Base54**(username**:**password)** |
| Authorization | Basic amF0aW46amF0aW4xMQ== |

**NOTE**: No matter, it is formLogin() or httpBasic(), after authentication verification, JSESSIONID is generated and further request verification is handled by JSESSIONID.

1. Let’s copy and paste from the WebSecurityConfigurerAdapter.configure(HttpSecurity) into our overriding method in ProjectSecurityConfig. configure(HttpSecurity http)
2. 
3. 