

User Controls



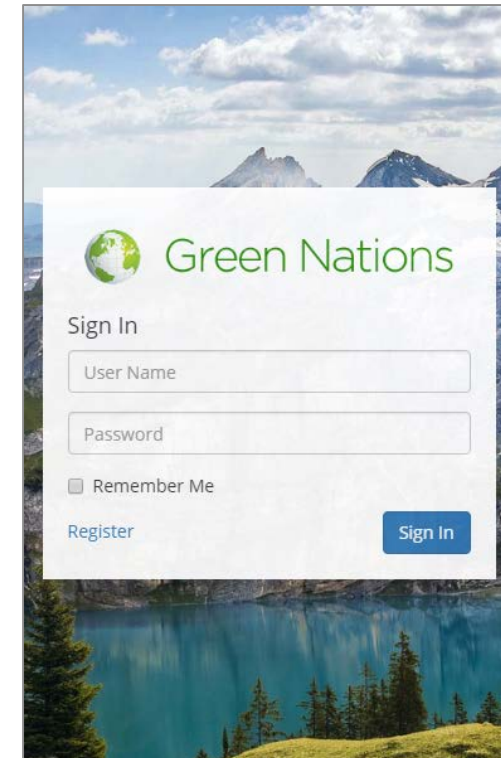
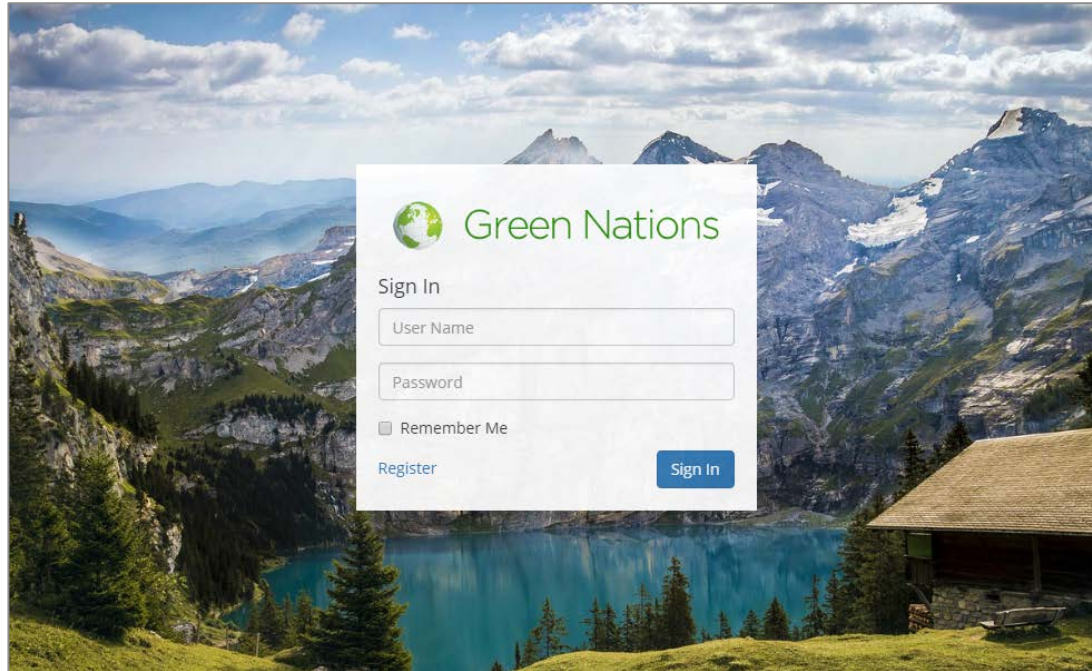
Mark Zamoyta
SOFTWARE DEVELOPER
@markzamoyta

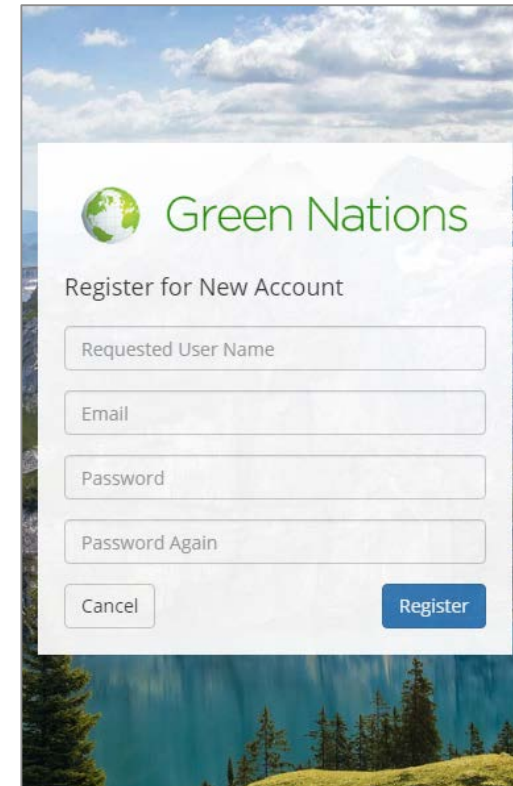
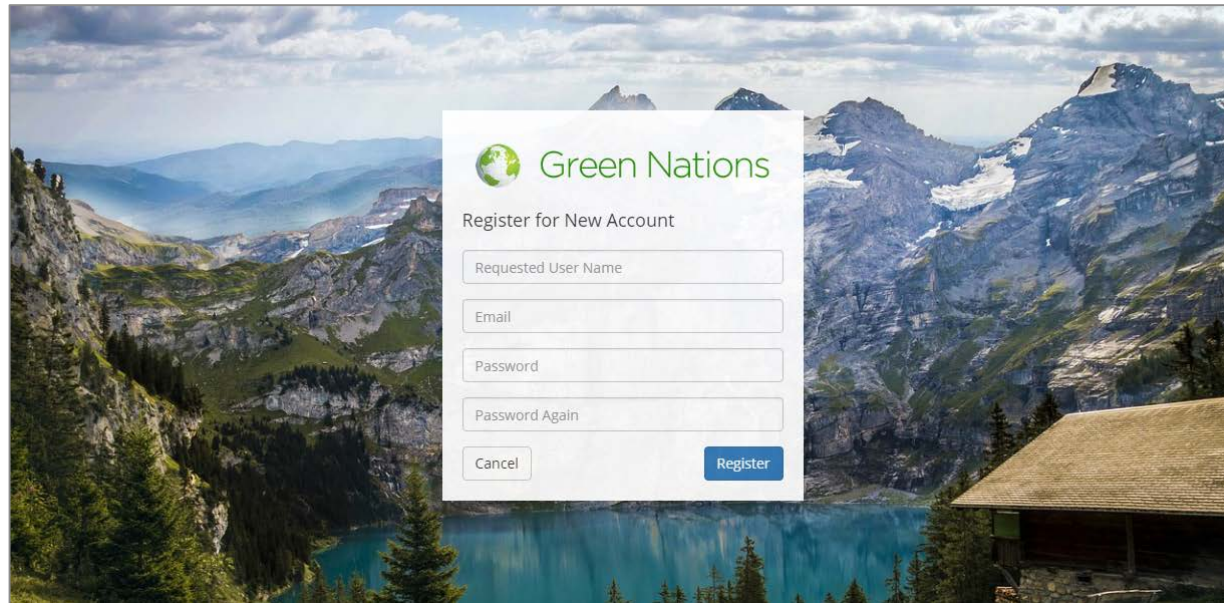




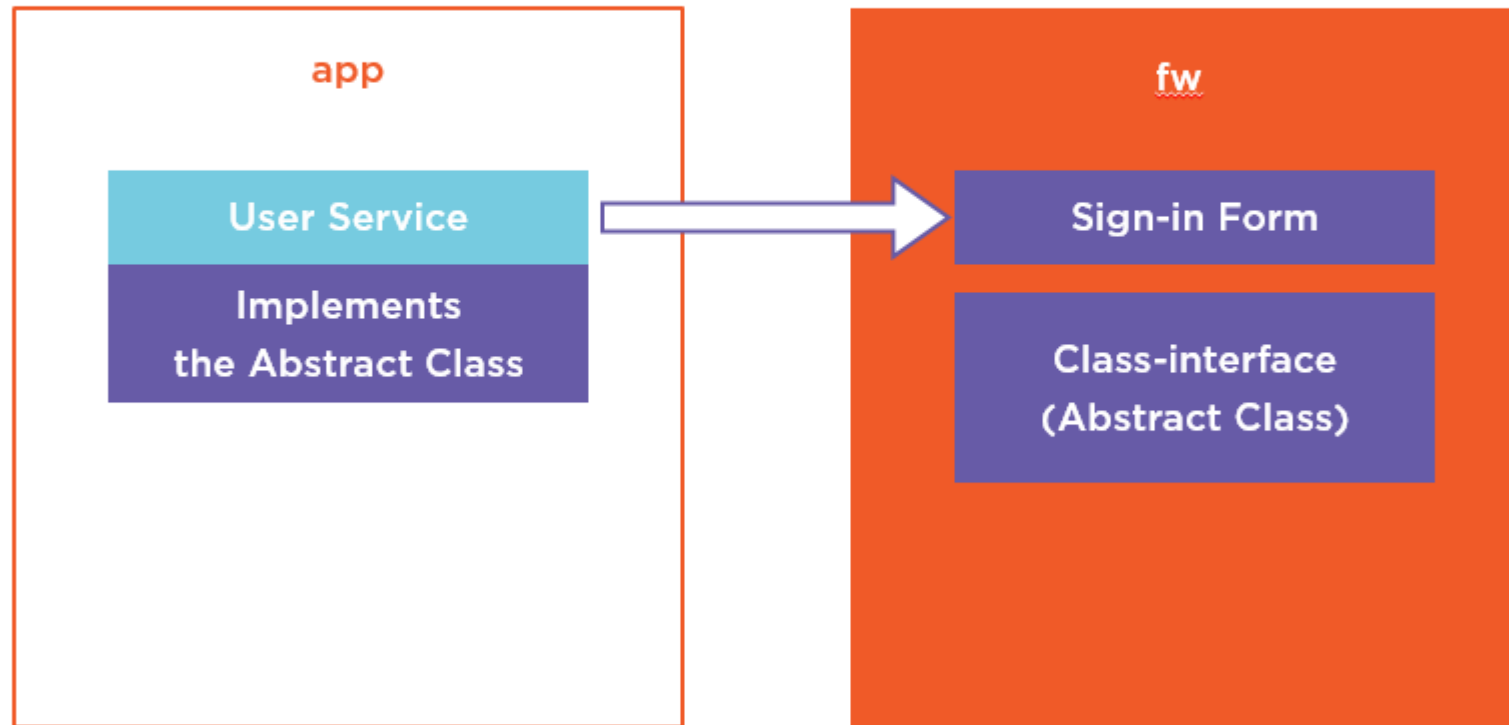
Module Overview

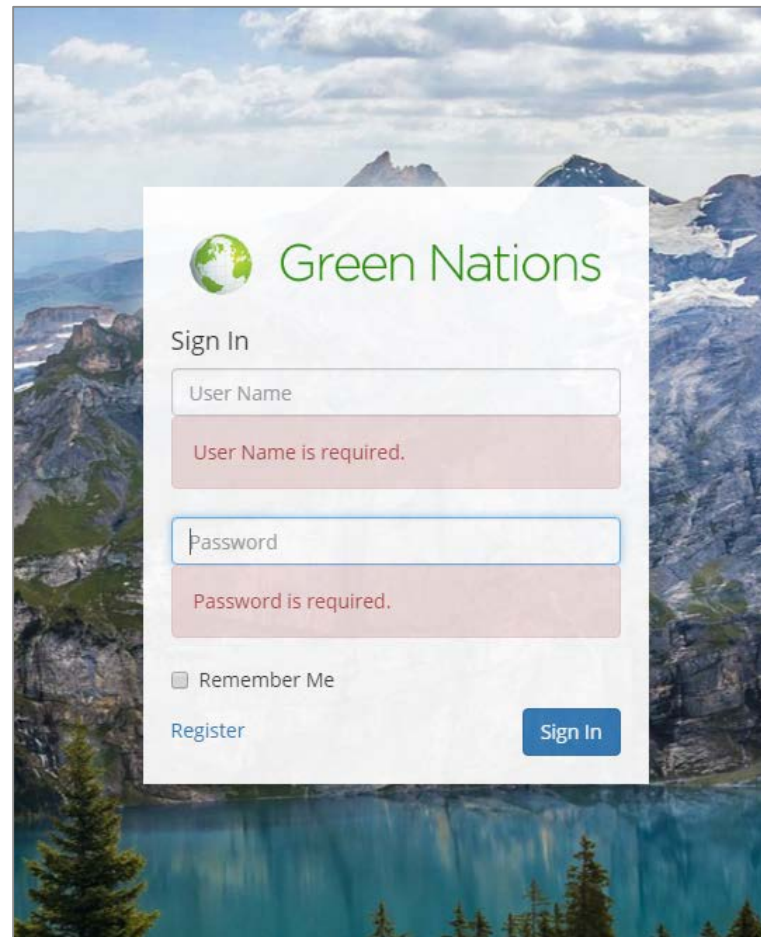







Using a Class-interface



A sign-in form for 'Green Nations' is centered on a scenic background of a mountain range and a lake. The form is white with a light blue border. At the top, there is a green globe icon followed by the text 'Green Nations'. Below this is the heading 'Sign In'. The form contains two input fields: 'User Name' and 'password'. The 'User Name' field has a red error message 'User Name is required.' below it. The 'password' field has a red error message 'Password is required.' below it. At the bottom left of the form is a checkbox labeled 'Remember Me'. At the bottom right is a blue button labeled 'Sign In'. A blue link labeled 'Register' is located at the bottom left of the form, below the 'Remember Me' checkbox.

 Green Nations

Sign In

User Name

User Name is required.

password

Password is required.

☐ Remember Me

[Register](#) [Sign In](#)






Green Nations



BAJAN ▾

Mark Zamoyta SIGN OUT

 DASHBOARD

 COUNTRIES ▾

 MAINTENANCE ▾



dashboard works!







Green Nations



BAJAN ▾

Mark Zamoyta SIGN OUT

 DASHBOARD

 COUNTRIES ▾

 MAINTENANCE ▾



dashboard works!



Routing and Child Routes



The Sign-in Component



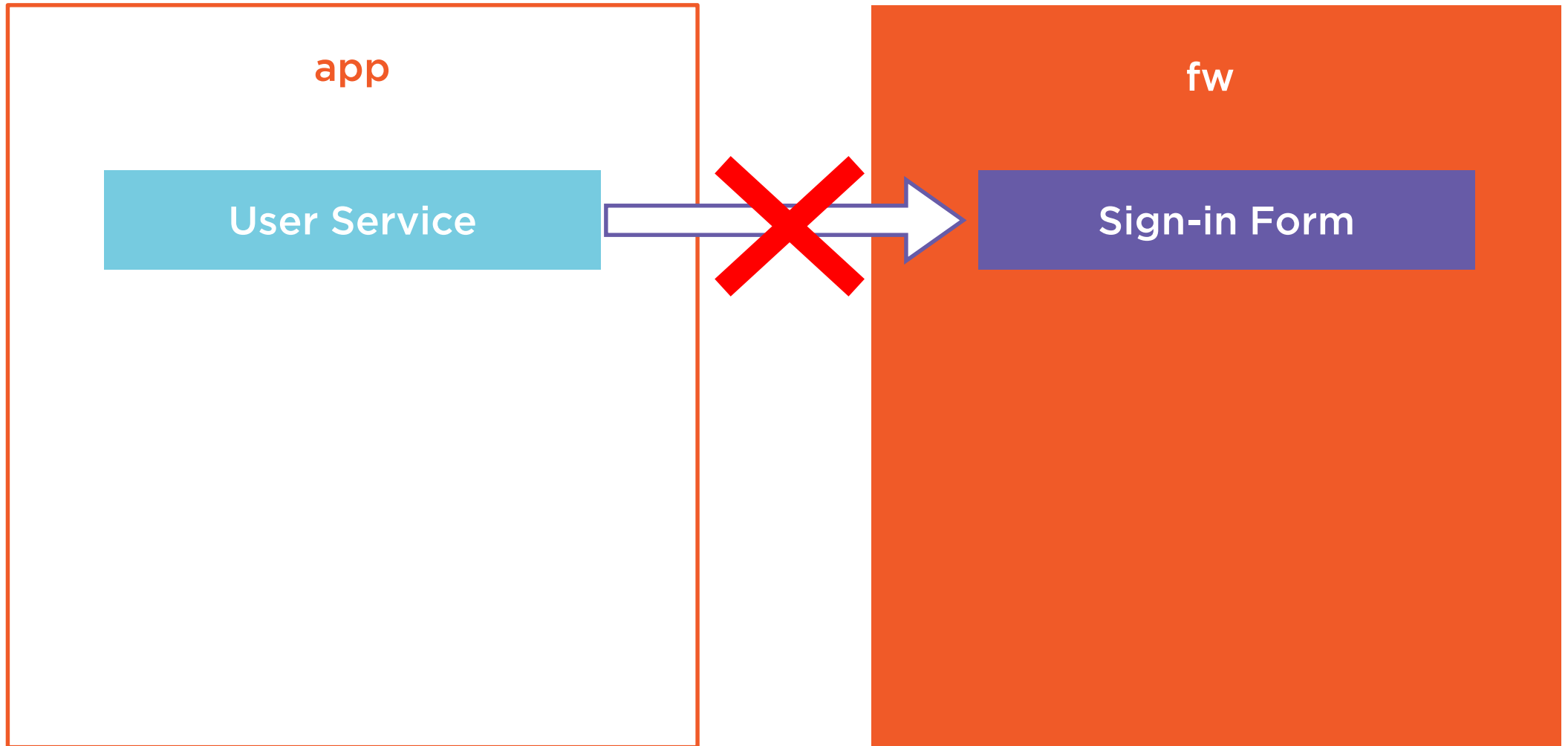
Building a Registration Form



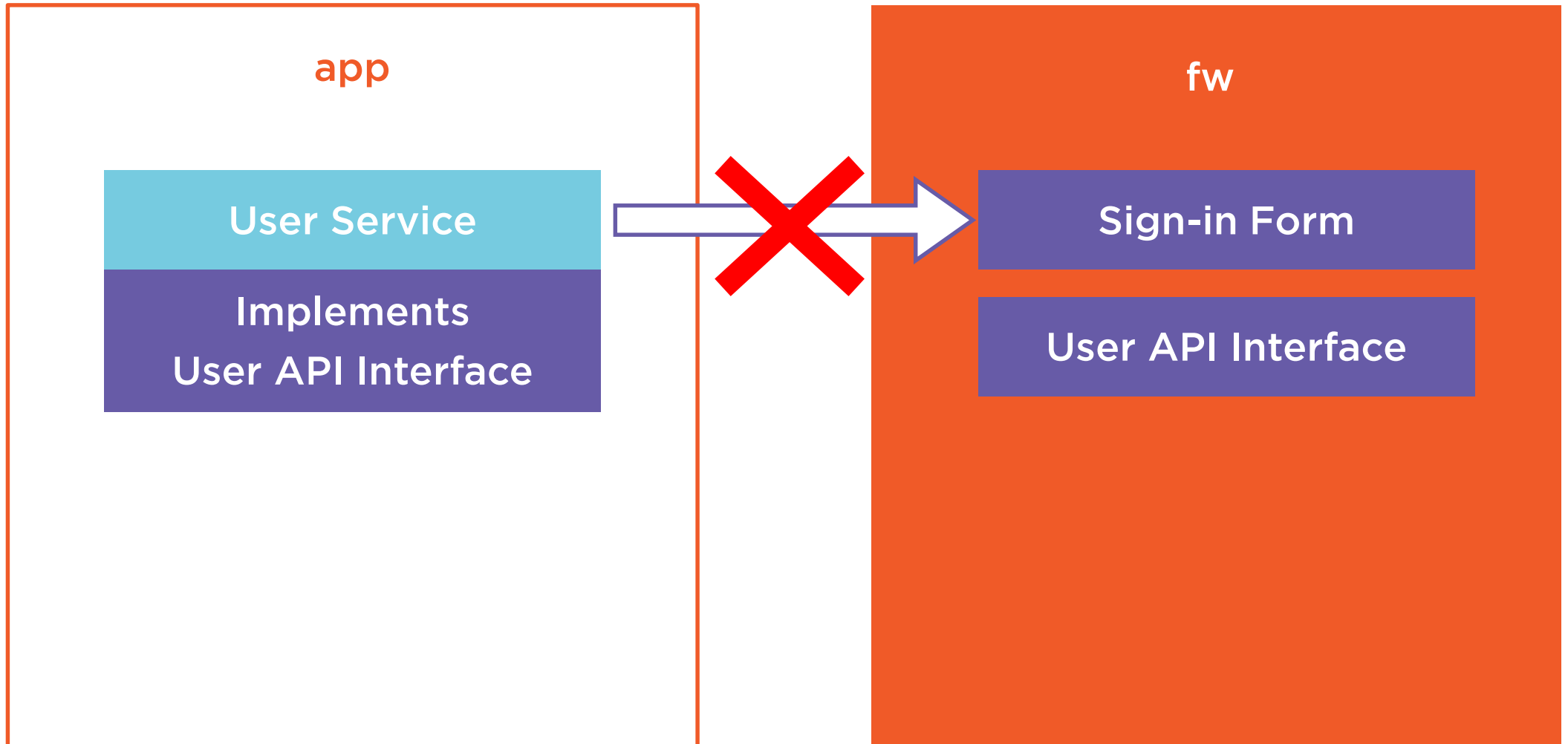
Dependency Injection and Class-interfaces



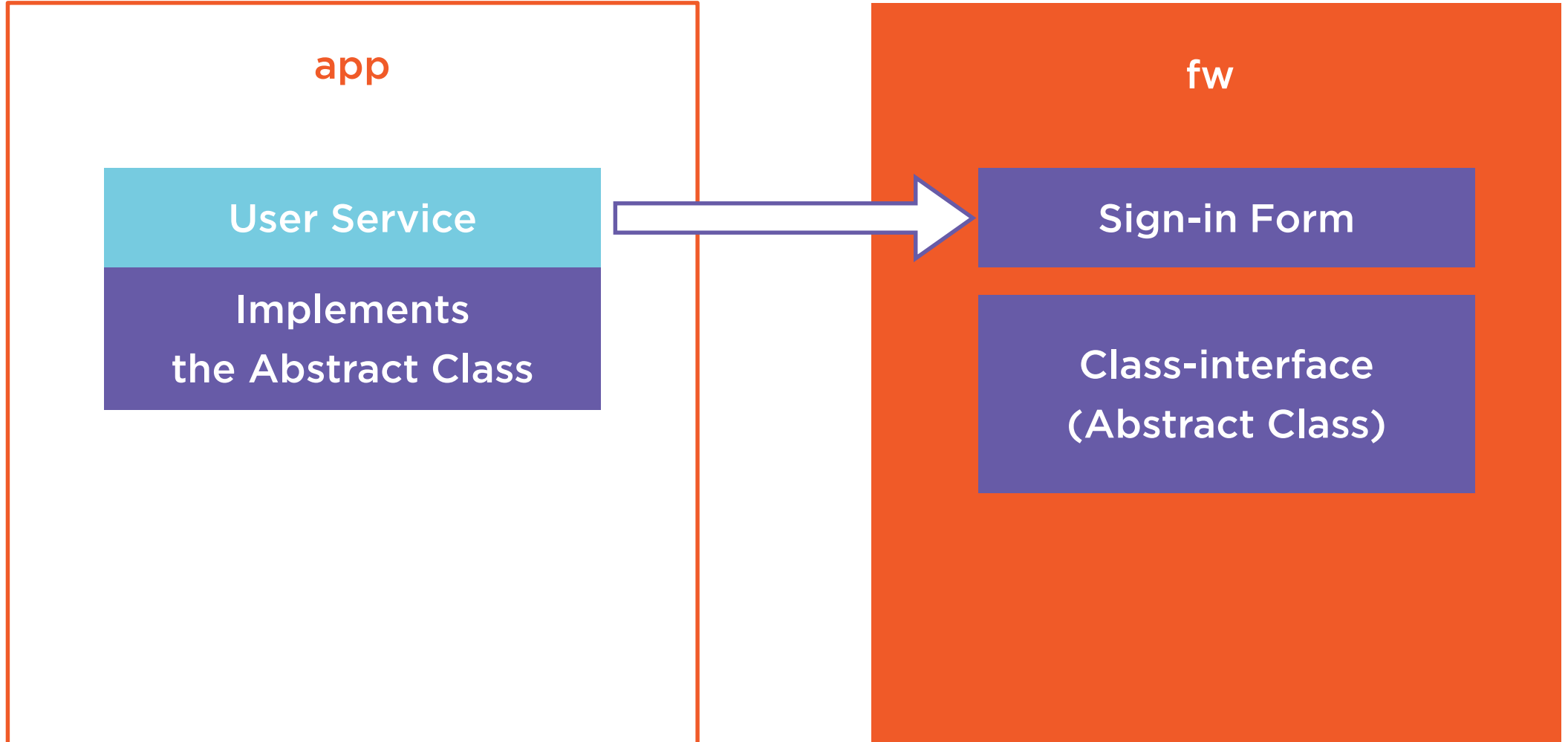
Using a Class-interface



Using a Class-interface



Using a Class-interface



Implementing a Class-interface



Dependency Injection Tokens



DI Using a Class-interface

```
providers: [  
    UserService  
]
```



DI Using a Class-interface

```
providers: [  
    // UserService  
]
```



DI Using a Class-interface

```
providers: [  
    // UserService  
    { provide: UserService, useClass: UserService }  
]
```



DI Using a Class-interface

```
providers: [  
  // UserService  
  { provide: UserService, useClass: UserService },  
  { provide: UserApi, useClass: UserService }  
]
```



DI Using a Class-interface

```
providers: [  
  // UserService  
  { provide: UserService, useClass: UserService },  
  { provide: UserApi, useExisting: UserService }  
]
```



DI Using a Class-interface

```
providers: [  
    UserService,  
    { provide: UserApi, useExisting: UserService }  
]
```



Submitting the Sign-in Form



Handling Sign-in Errors



Using Route Guards to Authorize Routes



Signing Out



Summary



Sign-in Form

Registration Form

Class-interface for DI

- implemented as an abstract class
- useExisting property for provider

Observables for Sign-in Form

Route Guards

Sign-out control

