



```
/*  
The following method builds a home page based off reccomended sites  
*/  
public WebPage buildLandingPage(PersonalizedReccomendationList userList)  
{ WebPage wb = new WebPage();  
  ReccomendationList list;  
  User user = LocalStorage.getInstance().getLoggedInUser();  
  if(user == null){  
    list = Database.getInstance().fetchDefaultReccomendations()  
  }  
  else{  
    list = userList;  
  }  
  wb.add(list);  
  return wb  
}
```


A character with long, wavy white hair and a crown with two pointed ears and cross-shaped ornaments. The character is wearing a white dress with a high collar and a dark bow at the waist. They are standing in a dark room with warm, orange-brown lighting. In the background, there are windows with metal frames. A large, colorful, abstract, and somewhat distorted overlay is positioned in front of the character's face and upper body. The text "Just Alias" is written in a large, white, bold, sans-serif font at the bottom of the image.

Just Alias

Alias Parameters

```
/*
The following method builds a home page based off recommended sites
*/
public WebPage buildLandingPage(PersonalizedReccomendationList userList)
{ WebPage wb = new WebPage();
  ReccomendationList list;
  User user = LocalStorage.getInstance().getLoggedInUser();
  if(user == null){
    list = Database.getInstance().fetchDefaultReccomendations()
  }
  else{
    list = userList;
  }
  wb.add(list);
  return wb
}
```



- An **alias parameter** is a **parameter** that is changed inside the method or different variants of a parameter are used depending on non-controllable conditional logic/factors.
 - Examples
 - The type of building permit returned may depend on a whether it exists in the current database
 - The browser engine may be dependent on our computer's architecture and system load.
- **Alias Parameters** prevent us from injecting and controlling our own mock data

Resolving Onion and Alias Parameters