

# API Application

\1 Internet Banking System\API Application



If you like PlantUML you may support us!  
<http://plantuml.com/patreon>



```
PlantUML 1.2023.10beta1
[From string (line 11) ]

@startuml

...
... ( skipping 212 lines )
...

hide stereotype
legend right
|<color:#000000>**Legend**</color> |
|<#08427B> person |
|<#1168BD> system |
|<#438DD5> container |
|<#686868> external person |
|<#999999> external system |
|<#B3B3B3> external container |
endlegend

rectangle "==Single Page Application\n//<size:12>[javascript and react]</size>\\n\n Provides all the internet banking f ..."
rectangle "==Mobile App\n//<size:12>[Xamarin]</size>\\n\n Provides a limited subset of the internet banking functionali ..."
rectangle "==Database\n//<size:12>[Relational Database Schema]</size>\\n\n Stores user registration information, hashed ..."
rectangle "==Mainframe Banking System\n\n Stores all of the core banking information about customers, accounts, transact ..."

rectangle "==API Application\n<size:12>[Container]</size>" <<boundary>> as api {
  Component(sign, "Sign In Controller", "MVC Rest Controlle", "Allows users to sign in to the internet banking system"...
  Syntax Error?
```



If you like PlantUML you may support us!  
<http://plantuml.com/patreon>



## Level 3: Component diagram

Next you can zoom in and decompose each container further to identify the major structural building blocks and their interactions.

The Component diagram shows how a container is made up of a number of "components", what each of those components are, their responsibilities and the technology/implementation details.

**Scope:** A single container.

**Primary elements:** Components within the container in scope. Supporting elements: Containers (within the software system in scope) plus people and software systems directly connected to the components.

**Intended audience:** Software architects and developers.