**逻辑层**

既用来描述我们步骤的逻辑，像点击，输入，选中，对比这些操作。

封装元素对应的操作以及输入数据

逻辑层基于关键字驱动。将对元素的操作根据口令存储，例如：

|  |  |  |  |
| --- | --- | --- | --- |
| Command | Target | OType | Value |
| Type.Input | Email\_Input | Target.sendKeys(Value) | Test data 1 |
| Type.Click | Login\_Button | Target,click() |  |

1. **<Function** command="LoginPage" target="" value=""**/>**
2. **<Function** command="sleep" target="5" value=""**/>**
3. **<Html** command="Type.Click" target="uimaps(LoginPage).targets.Login\_Button" value=""**/>**
4. **<Html** command="Type.Input" target="uimaps(LoginPage).targets.Email\_Input" value="Data(Data).LoginPage.Login\_Button\_1"**/>**

逻辑层总定义测试执行方式，所需要的对象和数据从对象层和数据层读取

**对象层(UIMaps)：**

用来存储被测试网页对象，按Page来保存文件，UIMaps.xml。

封装逻辑层中的Target

|  |  |
| --- | --- |
| 页面元素 | 元素XPATH |
| Login\_Button | //input[@id=’login’] |
| Email\_Input | //input[@id=’email’] |

1. **<LoginPage>**
2. **<Targets>**
3. **<xpath** name=”Login\_Button”**>**//input[@id=’login’]**</xpath>**
4. **<xpath** name=”Email\_Input”**>**//input[@id=’email’]**</xpath>**
5. **</targets>**
6. **</LoginPage>**

**数据层：**

封装逻辑层中的Value，Data.xml

1. **<Data>**
2. **<LoginPage>**
3. **<property** name=”Login\_Button\_1”**>**test data 1**</property>**
4. **<property** name=”Login\_Button\_2”**>**test data 2**</property>**
5. **</LoginPage>**
6. **<Data>**

===============================================================================

1、UIObject对象层：用来封装对象的操作方法(例：封装text field或textarea控件，实现输入功能)

2、构件层：主要用来描述页面上的控件，可以使用yml,xml,json甚至某种格式的文本文件进行描述，之后根据文件生成，这样可能会更加方面修改。

3、组件层：可以页面切分成大组件，然后对组件进行相关的操作，例如把Google的搜索作为一个组件。

4、断言层：这个部分用来执行testcase。