APIFORTRESS

UI Testing **Isn’t** API Testing

A series of obvious statements:

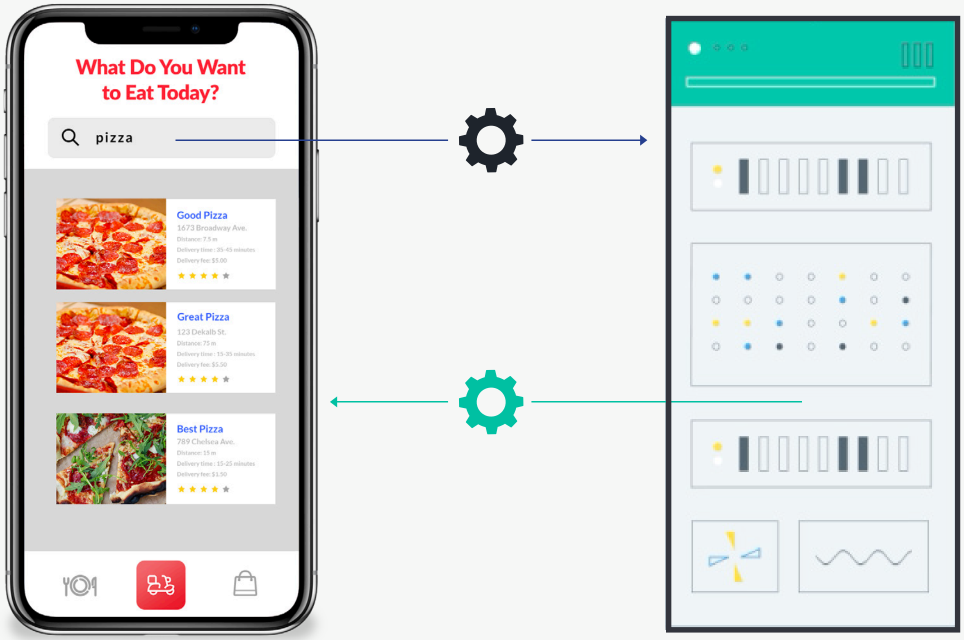
1. UI testing is important
2. Selenium and Appium are vital parts of automating that UI testing
3. Delivering new features or updates that don’t work properly is bad

What isn’t so obvious are the huge risks taken when people assume testing the UI will capture API bugs. First, let’s breakdown the importance of APIs. In this example let’s say we’re using a food delivery application:

Step 1.
User searches for local pizza options.

Step 2.
That search is sent to the app’s servers by API.

Step 3.
The app servers receives the API call, searches its database for pizza delivery options in the area.



```
{
“restaurants”: [
{
“id”:3456,
“name”: “Good Pizza”,
“distance”: 7.5,
“delivery_time”: “35-45”,
“delivery_fee”: 5,
“header_image”:
“https://food.com/img/3456/hero.png”
},
{
 “id”:3457,
“name”: “Great Pizza”,
“distance”: 75,
“delivery_time”: “15-35”,
“delivery_fee”: “5,5”,
“header_image”:
“https://food.com/img/3456/hero.png”
}
]
}
```

Step 6.
The app takes this data and converts it into a mobile friendly view.

Step 5.
That response is sent by API.

Step 4.
The server puts together a response in JSON.

Now here is an example of where problems can (and do) occur...

Did you notice anything wrong with the payload?

There are three errors in that JSON response, and it's unlikely any of them would be caught by an appium test.

```
"distance": 75,
```

First, Great Pizza is 75 miles away and shouldn't be in the list. Why was it listed? That can be any number of errors such as the app sent the wrong GPS location, the range wasn't set properly, or the platform doesn't properly confirm delivery range.

```
"delivery_fee": "5,5",
```

Second, Great Pizza's delivery fee of $5.5 has a comma in the middle. Commas are actually used instead of decimal points in many countries, and this data could have been uploaded by someone from Canada (for example).

```
"header_image":
"https://food.com/
img/3456/hero.png"
```

Finally, Great Pizza's hero image is the same as Good Pizza. If an API call is made asking for Great Pizza's hero image, then returning Good Pizza's hero is incorrect. A proper API test could validate the store ID of the request matches directory ID.

Forrester analyst, Diego Lo Giudice, looks at the rise in the popularity of API testing in the Forrester Wave in which he writes,

"Modern applications require a shift of the current 80% UI automation... shifting about 80% of that test automation [to] API test automation."

Giudice also writes, **"Modern applications are composed with APIs and services... [UI testing] cannot easily orchestrate tests that need to verify functional paths and back-end APIs and services."**

APIs are data. Data requires a different concept when it comes to testing, but not a complicated one. All information should be formatted correctly, accurate, within expected ranges, and as per what the original API designs were. Testing UI is imperative to delivering quality features and products, but ignoring the APIs that power those UIs with data and information is like only looking at the cover of a book before publishing.

Open the book and do some detailed proofreading with API Fortress.