

<epam> JDI 2.0 POWER OVERWHELMING



21 OCTOBER 2017

 **SEC**

ROMAN IOVLEV

Chief QA Automation

In Testing more than 12 years

In Testing Automation 10 years



roman.lovlev



roman_iovlev@epam.com



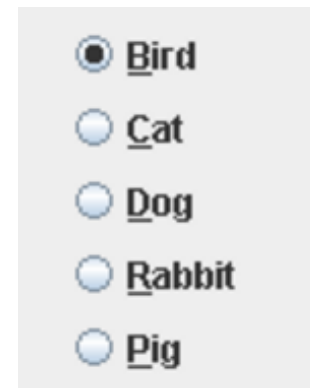
JDI?

UI Test Automation Framework



- UI Test Framework
- UI Elements oriented
 - Dozens of UI elements already implemented
 - Most of common problems already solved (e.g. stabilization)

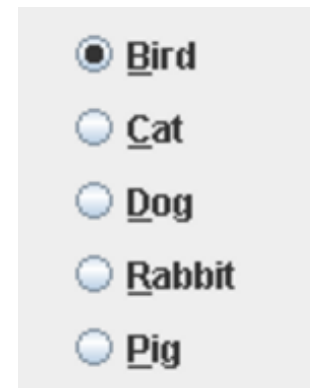
- UI Test Framework
- UI Elements oriented



An 'Order product' form with the following fields: Product (Тестова статийка), Quantity (with a red asterisk), Your name (with a red asterisk), and Your e-mail (with a red asterisk). The form is powered by Highslide JS.

Senior ATG Developer	Software Engineering	San Francisco, CA, USA
Front-End Engineer	Software Engineering	Mountain View, CA, USA
Senior JavaScript Developer	Software Engineering	USA

- UI Test Framework
- UI Elements oriented



powered by highslide.js

Order product

Product
Тестова статийка

This is a silly test article. Who would ever think of buying it?...

Quantity: *

Your name: *

Your e-mail: *

Senior ATG Developer 🔥	Software Engineering	San Francisco, CA, USA
Front-End Engineer 🔥	Software Engineering	Mountain View, CA, USA
Senior JavaScript Developer 🔥	Software Engineering	USA

- Interfaces above engines



- Save your time for Automation Testing
 - Provide stable and detailed results
 - Fast on start and easy for support tests in future
 - Platform - Flexible for any UI project
- JDI learn good automation practices

- Write test code **faster up to 5 times**
 - Average result around 2.8 times
 - 4.7 times speedup on the project with standard implementation
- Produce **less** amount of **test code (loc) up to 3 times**
 - Average result around 2.1 times
 - 2.8 times reduction on the project with standard implementation
- Achieve **higher clearness of tests**
 - Decrease of support time for test projects
 - Lowering of project entry barrier for newcomers
- Complete projects with **higher quality**
 - Based on 70 % answers in survey



- **Reuse investments** from one Project on another
 - Based on 5 years of work and more than 30 projects that already use JDI
- **Save up to 80% test effort** by migrating tests to other Platforms
 - Based estimated average scope reductions for all test process stages
 - Example: migrate Web tests to Mobile platform
- Can be **used in most of projects** with UI Automation
 - Actually we have no projects where JDI is not applicable. The only reason why not all of our projects use JDI is Client requirements
- **Save up to 30-40% money** from whole test process
 - Based on average calculation of scope reductions for all test process stages

JDI 2.0



WEB



MOBILE



DESKTOP

- Less code for new Engines
- More unification
- Guideline for Engines developers
 - Implement list of methods with your engine
 - Add your engine specific features
 - Create list of elements (add only specific functionality)



- Elements collections
 - Simple Web
 - Extended Web
 - Complex forms and tables
 - Shop elements
 - Bank / Government / Document elements

...Your collections

- All elements and methods documentation
 - Simple start guideline
 - Cookbook: JDI features tutorial
 - Guideline for contributors
- ... anything else?





JDI 2.0 FEATURES

- Implemented UI Elements
- UI Elements based PageObjects with cascade initialization (`WebSite.init(EpamSite.class)`)
- Cascade locators inheritance
- No more sleeps or waits
- Implemented complex elements (Form, Table)
- Automatic Latest Driver download
- Customizable elements and actions

- Business actions detailed log with no effort
- Native integration with reporting (Allure, RP)
- Cross language/version tests
- Enums based actions
- Entity Driven Testing native support
- File based properties
- Customizable page loading and get element strategies

- ❖ Verify layout
- ❖ PageObjects generator as Library
- ❖ Cucumber tests manager
- ❖ Selenium-Selenide integration
- ❖ JDI Http
- ❖ JDI on Python
- ❖ Angular support...



```
@Image("/src/test/resources/submitbtn.png")  
@FindBy(text = "Submit")  
public Button submit;
```

```
@Image("/src/test/resources/submitbtn.png")
```

```
@FindBy(text = "Submit")
```

```
public Button submit;
```

```
submit.isDisplayed();
```

```
submit.assertDisplayed();
```

```
@ImageFolder("/src/test/resources/imgs")
```

```
public EpamSite extends WebSite;
```

```
@Image("submitbtn.png")
```

```
@FindBy(text = "Submit")
```

```
public Button submit;
```

```
public class EpamSite extends WebSite {
    public static HomePage homePage;
    public class HomePage extends WebPage
        @FindBy(text = "Submit")
        public Button submit;
```

```
“src/test/resources/jdi-images/epamsite/
homepage/submit.jpg”
```



```
public class EpamSite extends WebSite {
    public static HomePage homePage;
    public class HomePage extends WebPage
        @FindBy(text = "Submit")
        public Button submit;

    homePage.verifyLayout()
    homePage.assertLayout() / homePage.checkLayout()
```

RULES

```
{"elements": [{  
  "type": "Button",  
  "name": "value",  
  "css": "input[type=button]"  
},  
  ...  
}]}
```

URLS

```
https://domain.com/  
https://domain.com/login  
https://domain.com/shop  
https://domain.com/about
```

OUTPUT

src/main/java

PACKAGE

com.domain

```
new PageObjectsGenerator(rules, urls, output, package)  
    .generatePageObjects();
```

```
<input type="button" value="Next">  
<input type="button" value="Previous">  
<button class="btn">Submit</button>
```

```
"type": "Button",  
"name": "value",  
"css": "input[type=button]"  
  
"type": "Button",  
"name": "text",  
"css": "button.btn"
```

```
@FindBy(css="input[type=button][value=Next]")  
public Button next;  
  
@FindBy(css="input[type=button][value=Previous]")  
public Button previous;  
  
@FindBy(xpath="//button[@class='btn'  
and text()='Submit']")  
public Button submit;
```

Bootstrap rules

Html 5 rules

Telerik rules

KendoUI rule

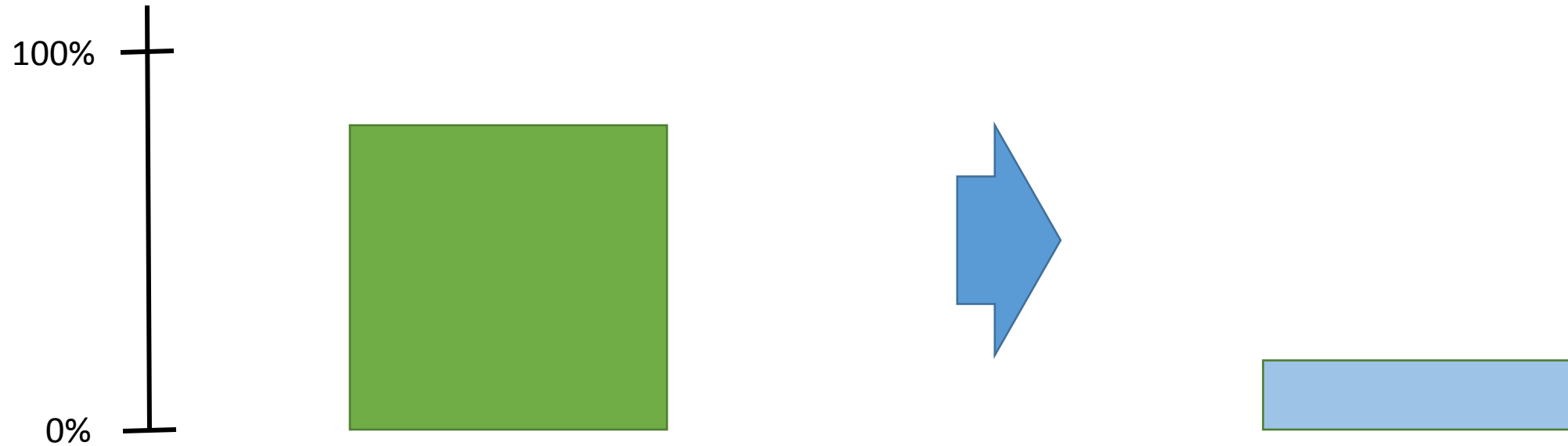
...

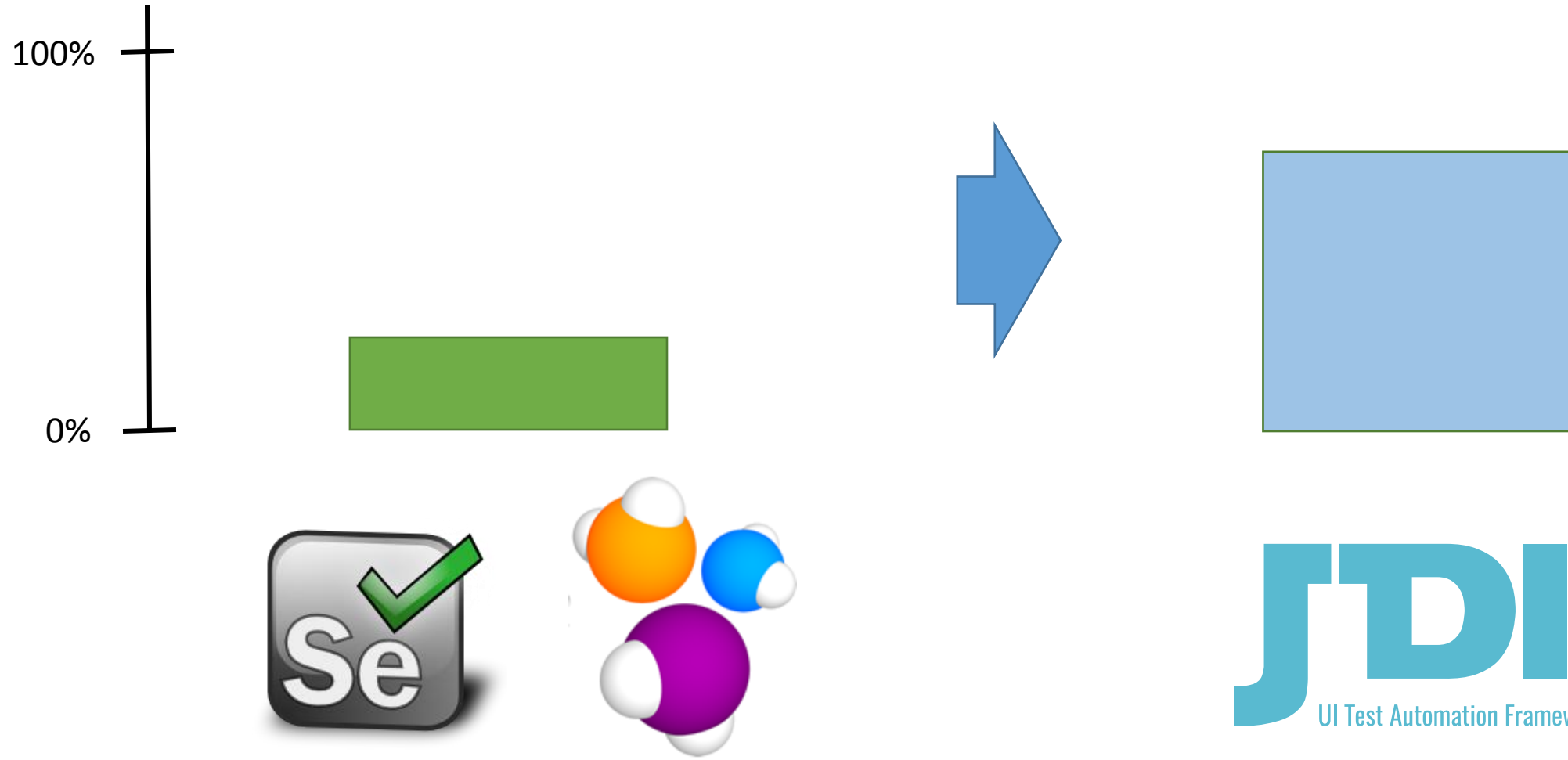
Your awesome UI framework rules

SELENIUM-SELENIDE INTEGRATION



SELENIUM-SELENIDE INTEGRATION






```
@ServiceDomain("http://httpbin.org/")  
public class ServiceExample {  
    @GET("/get") static RestMethod getMethod;  
    @POST("/post") RestMethod postMethod;  
    @PUT("/put") RestMethod putMethod;  
    @PATCH("/patch") RestMethod patch;  
    @DELETE("/delete") RestMethod delete;
```

```
RestResponse resp = ServiceExample.getMethod.call();  
assertEquals(resp.status, 200);  
assertEquals(resp.statusType, OK);  
assertEquals(resp.jsonBody("url"),  
"http://httpbin.org/get");  
assertEquals(resp.jsonBody("headers.Host"),  
"httpbin.org");
```



Angular

`@JFindBy (model="btn-model")`

`@JFindBy (binding="btn")`

`@JFindBy (repeat="r-button")`

Special

`@JFindBy (text="Submit")`

`@JFindBy (name="Submit")`

`@JFindBy (value="Submit")`

`@JFindBy (title="Submit")`

`@JFindBy (type="Submit")`

Selenium

`@JFindBy (css=".submit")`

`@JFindBy (tagName="button")`

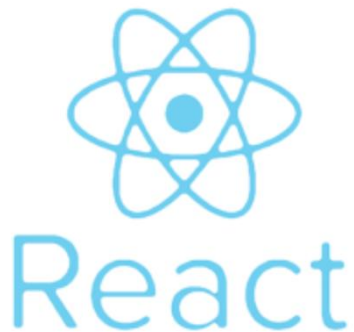
`@JFindBy (linkText="Test")`

`@JFindBy (partialLinkText="st")`

`@JFindBy (xpath="//button")`

`@JFindBy (id="submt-id")`

`@JFindBy (className="submit")`





Drag & drop your resume

Message/cover letter/link to your CV
(optional)

☒ Subscribe to job updates

By checking this box, you agree to receive job promotions, events and other important news.

0aedd6



Enter security code

CANCEL

SUBMIT →

```
class AddCVForm(Form):
    first_name = TextField (By.css("[placeholder='First Name']"))
    last_name = TextField (By.css("[placeholder='Last Name']"))
    email = TextField (By.css("[placeholder='Email']"))
    country = Dropdown (root=By.css(".country-wrapper"),
                        by_select_locator=By.css(".arrow"),
                        ...)
    city = ...
    cv = RFileInput (By.css(".file-upload"))
    comment = TextArea (By.xpath(".comment-input"))
    submit = Button (By.xpath("//*['Submit']"))
    cancel = Button (By.xpath("//*['Cancel']"))
```

<https://goo.gl/mvcWfc>



<https://github.com/epam/JDI>



<http://jdi.epam.com/>



https://vk.com/jdi_framework



README

