



JDI SETTINGS

20 SEP 2016

- UI Objects. JDI
- JDI. Composite elements
- **JDI Settings**
- JDI for any UI

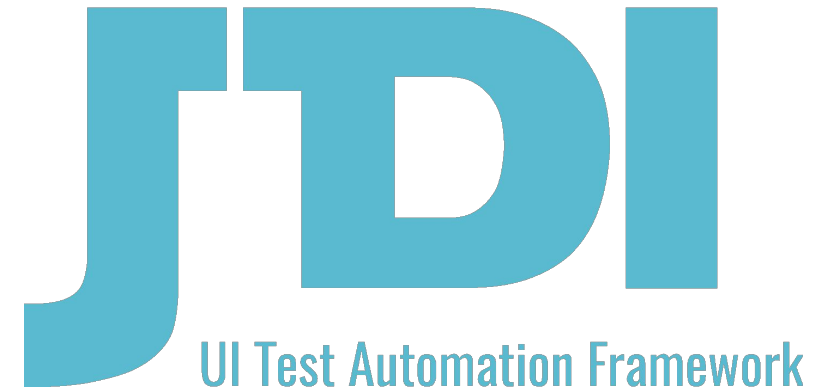
- **Test project from scratch**
- **Test settings**
- **Customization**
- **Preconditions**
- **Parallel run**
- **Logging**
- **Enums**
- **Matchers**



NEW PROJECT

Maven, Gradle, Ivy

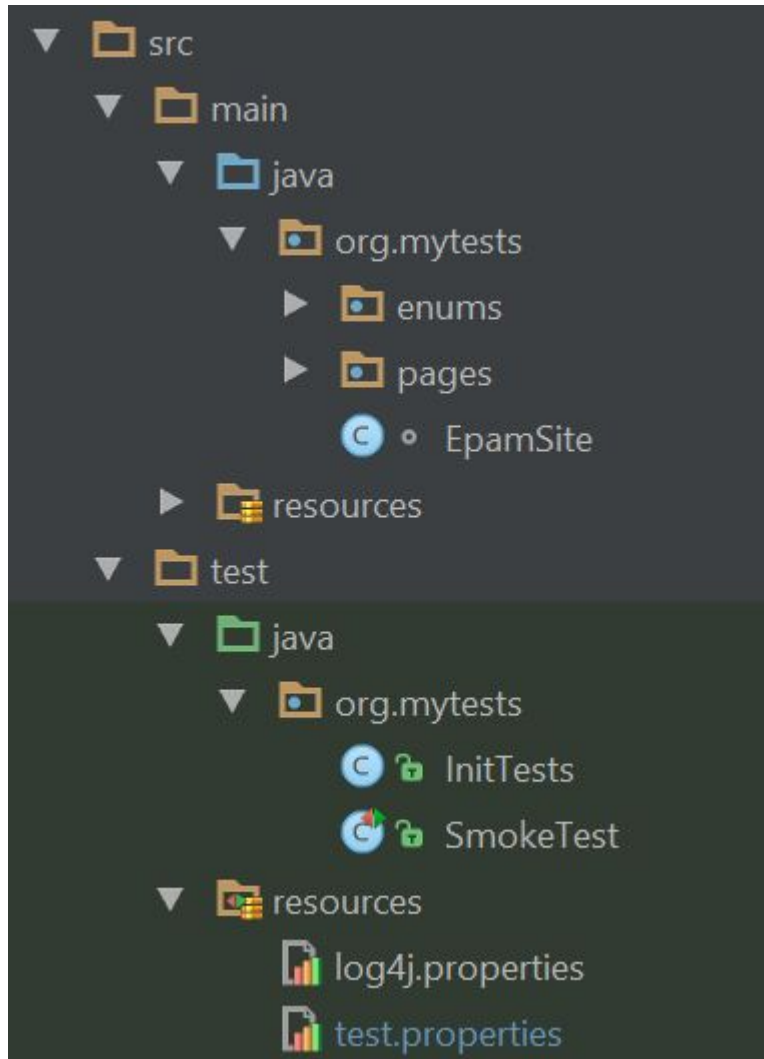
```
<dependency>  
  <groupId>com.epam.jdi</groupId>  
  <artifactId>jdi-uitest-web</artifactId>  
  <version>1.0.67</version>  
</dependency>
```



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

DEMO: TEST PROJECT FROM SCRATCH





- test.properties
- log properties
- ui objects
- test init
- tests

```
driver=chrome
timeout.wait.element=10
domain=https://www.epam.com/
driver.getLatest=true
search.element.strategy=strict | soft
browser.size=1800X1000
demo.mode=false | true
#multithread=true
#run.type=local | remote
screenshot.strategy=on fail | on | off
```


Log4J

log4j.rootLogger = **info**, **console**

log4j.appender.**console** = org.apache.log4j.ConsoleAppender

log4j.appender.**console**.layout = org.apache.log4j.PatternLayout

log4j.appender.**console**.layout.ConversionPattern = **%m%n**

log4j.rootLogger = **debug|error**, **file**, **HTML**, **dailylog**

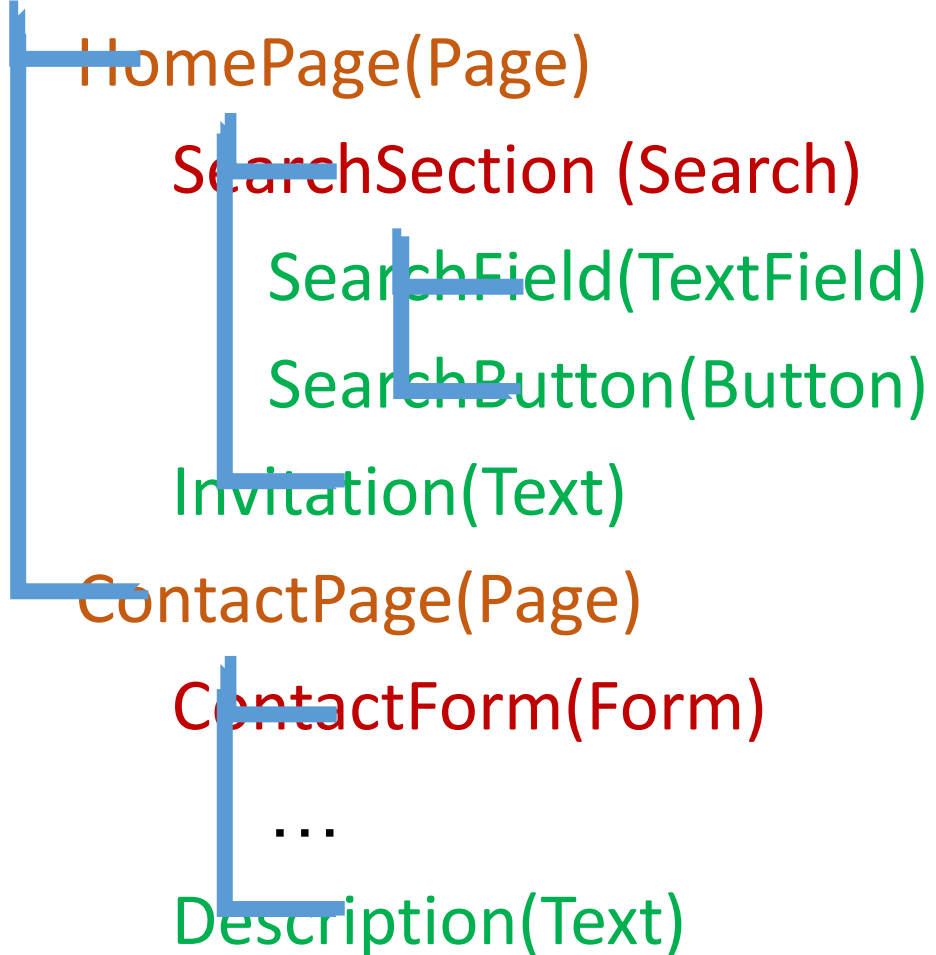
log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=target/.logs/events.log

log4j.appender.file.layout.ConversionPattern= %d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L -
%m%n

log4j.appender.file.layout=org.apache.log4j.PatternLayout

EpamSite (Site)



```
@BeforeSuite(alwaysRun = true)
public static void setUp() {
    WebSite.init(EpamSite.class);
}
```



```
public Dropdown<Types> productTypes
    = new Dropdown<Types>() {
    @Override
    public void selectAction(String name) {
        super.selectAction(name);
        label.click();
    }
};
```

```
public class TreeDropdown<T extends Enum> extends Dropdown<T> {  
    ...  
    @Override  
    protected void selectAction(String name) {  
        expandAction();  
        String[] nodes = name.split(" > ");  
        SearchContext context = getDriver();  
        if (treeLocators.size() >= nodes.length)  
            for (int i=0; i < nodes.length; i++) {  
                String value = nodes[i];  
                context = first(context.findElements(treeLocators.get(i)),  
                    el -> el.getText().equals(value));  
                new Clickable((WebElement) context).click();  
            }  
        }  
    }  
}
```


@BeforeSuite (alwaysRun = true)

```
public static void setUp() {
```

```
    ActionScenrios.actionScenario = (element, actionName, jAction, level) -> {
```

```
        logger.info(format("Do '%s' action", actionName));
```

```
        jAction.invoke();
```

```
    };
```

```
    ActionScenrios.resultScenario = (element, actionName, jAction, logResult, level) -> {
```

```
        logger.debug(format("Do '%s' action", actionName));
```

```
        Object result = jAction.get();
```

```
        logger.info(format("Get '%s' action result: %s", actionName, result));
```

```
        return result;
```

```
    };
```

```
}
```

```
@BeforeSuite (alwaysRun = true)
public static void setUp() {
    SeleniumDriverFactory.elementSearchCriteria =
        el -> el.isEnabled() && el.isDisplayed();
    //    el -> el != null;
};

@Test()
public void simpleTest() {
    sumResult.avatar.localElementSearchCriteria
        = el -> el.getText().equals("5");
};
```



~~`@Test(dependsOnMethods = "loginTest")`~~

~~`public void simpleTest() {`~~

~~`...`~~

~~`};`~~

~~`@Test(dependsOnGroups = "smoke")`~~

~~`public void simpleTest() {`~~

~~`...`~~

~~`};`~~

1. Independent tests
2. Time optimization

PRECONDITION

1. IsInStateCheckAction
2. MoveToStateAction

JDI Page precondition

homePage.isOpened();

JDI State precondition

PreconditionsState.isInState(LOGGED_IN)
or
isInState(LOGGED_IN)


```
public enum Preconditions implements IPreconditions {  
    CALC_INIT(() -> calculator.value == 0,  
        () -> { calculator.clear();  
                calculator.clearMemeory() } ),  
    CAREERS_PAGE("/careers");  
    ....  
}
```





USING ENUMS

```
public Menu<Options> topMenu;
```

```
public enum Options {  
    Home, About, Contacts }
```

```
topMenu.select(About$.About);
```

```
public Dropdown<Colors> colors;
```

```
public Tabs<Areas> areas;
```

```
public Checklist<Settings> settings;
```

```
public ComboBox<Tags> tags;
```

```
public DropList<Sizes> shirtSizes;
```

```
public Selector<VoteOptions> vote;
```

```
public RadioButtons<Ratings> rating;
```

```
public enum Options {  
    Home('option-1'),  
    About('option-3'),
```

```
    public String value;  
    Options (String value) { this.value = value; }  
    @Override  
    public String toString() { return value; }  
}
```



LOGGER

Log4J

log4j.rootLogger = **info**, **console**

log4j.appender.**console** = org.apache.log4j.ConsoleAppender

log4j.appender.**console**.layout = org.apache.log4j.PatternLayout

log4j.appender.**console**.layout.ConversionPattern = **%m%n**

log4j.rootLogger = **debug|error**, **file**, **HTML**, **dailylog**

log4j.appender.file=org.apache.log4j.RollingFileAppender

log4j.appender.file.File=target/.logs/events.log

log4j.appender.file.layout.ConversionPattern= %d{yyyy-MM-dd HH:mm:ss} %-5p %c{1}:%L -
%m%n

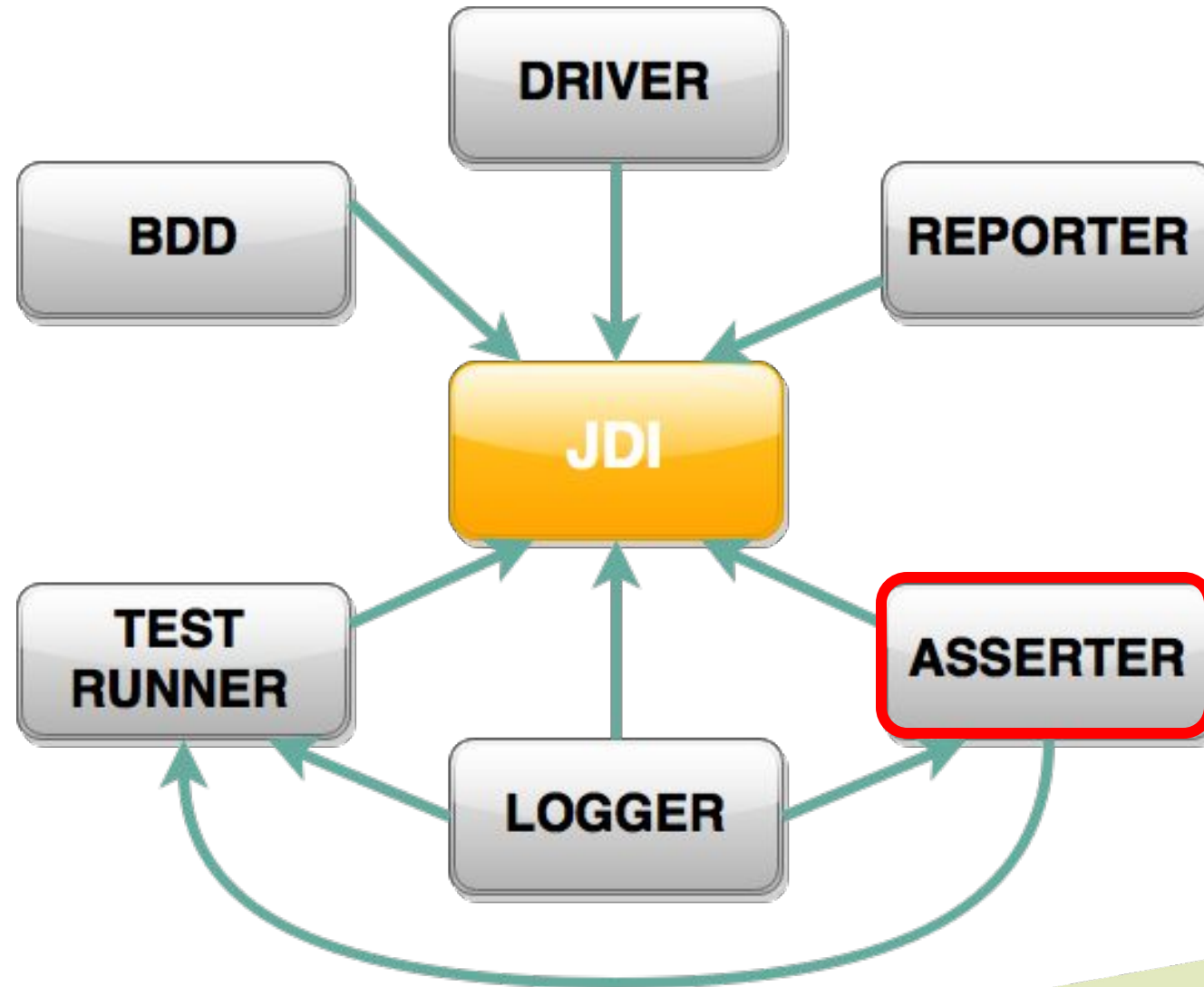
log4j.appender.file.layout=org.apache.log4j.PatternLayout

```
JDISettings.logger = SuperLogger.Logger();
```

```
logger.info("Start tests");
```

```
||  
|| [Info] 10:20.154 Start tests  
||  
|| [Info] 10:20.220 Select Colors' for Selector 'Selector' (.Selector; css='.colors')  
||  
|| [Debug] 10:21.004 Get web element for Clickable 'Clickable' (.Clickable; css='.colors')  
||  
|| [Info] 10:21.932 Done  
||
```







MATCHERS


```
Assert.contains("Test Text", "Text");  
Assert.matches("1352-423-85746", "\\d{4}-\\d{3}-\\d{5}");  
  
Assert.arrayEquals(searchResults, expectedResults);  
Assert.listEquals(orders, expectedOrders);  
Assert.each(searchResults).contains("iPhone");  
Assert.each(searchResults).matches("iPhone \\d.*");  
  
Assert.areEquals(() -> getNext(), "iPhone 6");  
Assert.contains(() -> getNext(), "iPhone 5")  
  
Assert.throwException(this::request, "Bad Request");  
Assert.hasNoExceptions(this:: request);
```

```
new Check("Search results are correct")  
    .listEquals(searchResults, expectedResults);
```

```
ScreenAssert.matches("1352-423-85746", "\\d{4}-\\d{3}-\\d{5}");
```

```
Assert.ignoreCase().areEquals(result, "iPhone 6");
```

```
Assert.waitTimeout(2).contains(() -> result, "iPhone");
```

```
Assert.doScreenshot(SCREEN_ON_FAIL).isTrue(2 * 2 == 4);
```

```
Assert.fail("Houston we have a problem");
```





```
throw Assert.exception("Something goes wrong");
```



ENTITY TABLES





```
@JTable(  
    root = @FindBy (css = "offers"),  
    row = @FindBy (xpath = ".//li[%s]//div"),  
    column = @FindBy (xpath = ".//li//div[%s]"),  
    header = {"ID", "Title", "Apply"}  
    rowsHeader = {"1", "2", "3", "4"},  
    headers = @FindBy(xpath = "//tr"),  
    rowNames = @FindBy(xpath = "//td"),  
    cell = @FindBy(xpath = "//li[{1}]/div[{0}]"),  
    footer = @FindBy(xpath = "//footer"))  
public Table Offers;
```

```
@JTable(  
    height = 4,  
    width = 3,  
    size = "3x4",  
    rowStartIndex = 1,  
    colStartIndex = 1,  
    headerType = COLUMNS_HEADERS,  
    useCache = false)  
public Table Offers;
```

#	Type	User Name	Description
1	ИП ▼	Иван Иванович	 Описание задачи <input type="checkbox"/> Выбрано
2	Физ.Лицо ▼	Сергей Сергеевич	 Описание задачи <input type="checkbox"/> Выбрано
3	Юр.Лицо ▼	Роман Юрьевич	 Описание задачи <input type="checkbox"/> Выбрано
			 Lorem ipsum

```

@FindBy(id = "user-table")
public EntityTable<Entity, Row> Users =
    new EntityTable<>(Entity.class, Row.class);
  
```

#	Nype	User Name	Description
1	ИП ▼	Иван Иванович	 Описание задачи <input type="checkbox"/> Выбрано
2	Физ.Лицо ▼	Сергей Сергеевич	 Описание задачи <input type="checkbox"/> Выбрано
3	Юр.Лицо ▼	Роман Юрьевич	 Описание задачи <input type="checkbox"/> Выбрано
			 Lorem ipsum

@Data

```
public class Entity {
    public String number;
    public String type;
    public String
userName;
    public String
description;
}
```

```
public class Row {
    public Text number;
    public Dropdown type;
    public Link userName;
    public Description
description;
}
```



```
Row user = Users.firstRow(r ->  
    textOf(r.description).contains("задачи") &&  
    textOf(r.type).equals("ИП"));
```

↓ Column

↓ Checkbox

```
user.Description.Select.select();
```

```
List<Entity> usersList = Users.entites();
```

```
Assert.entitiesAreEquals(usersList, ЭталонныйПользователь);
```



- UI Objects. JDI
- JDI. Composite elements
- JDI Settings
- **JDI for any UI**



roman.iovlev



roman_iovlev@epam.com