Creating a new execution engine

When creating an execution engine, you are required to implement three abstract methods in the Subclass of ExecutionEngine (see JavaDoc for more details). These abstract methods are as follows

```
        public abstract class ExecutionEngine {
                ...
                public abstract void init(Element element);
                public abstract void execute(Element element);
                public abstract void onClose();
                ...
        }
```

API developers can insert any desired code into the above abstract methods and get the events when the execution engine is initialized, executed, and terminated.
You can execute the following code to create a MyExecutionEngine class

```
        public class MyExecutionEngine extends ExecutionEngine {

                public MyExecutionEngine(ExecutionEngineDescriptor engineDescriptor) {
                        super(engineDescriptor);
                }

                @Override
                public void execute(Element element) {
                ...
                }

                @Override
                public void init(Element element) {
                        Project project = Project.getProject(element);
                        // add engine listener if needed
                        EngineListener listener = new MyEngineListener(project);
                        addEngineListener(listener);
                }

                @Override
                public void onClose() {
                ...
                }
        }
```

The created class MyExecutionEngine must be returned from createEngine() of its descriptor as defined in Creating an execution engine descriptor.