

## Tutorial 1 – HelloMachine

This is the first tutorial which will show you how to write your first C<sup>3</sup> component. As you will see it is really quick and easy even without code generation.

For the first component, tradition continues by creating a HelloMachine component, the user of the component in this case a main program will display the famous hello world to the standard output.

The tutorial will be divided into two clear sections. The first one defines the component's contract which contributes to define and describe the behaviour of the component. The second one is the implementation which provides the code which can be executed following the contract.

### Create the Contract

First create a package with the name of component, java naming rule has to be applied therefore the name of the package shall be in lowercase.

### Name the component

First a component shall be named. To do that create a new Interface with the name of the component. The new interface shall also extend *org.capcaval.c3.Component*.

```
package org.capcaval.c3.sample.tutorial1.hellomachine;  
  
import org.capcaval.c3.component.Component;  
  
public interface HelloMachine extends Component{  
}
```

### Define the service

Service can be compared to the method for object. In this sample, the service is building a sentence to salute a given name.

```
package org.capcaval.c3.sample.tutorial1.hellomachine;  
  
import org.capcaval.c3.component.ComponentService;  
  
public interface HelloMachineServices extends ComponentService{  
    public String salute(final String name);  
}
```

That all for the contract, we know what the component do.

## Implementation of the contract

This phase is very simple you have just to implement the contract. In our case the name is put inside a salute sentence.

```
package org.capcaval.c3.sample.tutorial1.hellomachine.impl;
import org.capcaval.c3.sample.tutorial1.hellomachine.HelloMachine;
import org.capcaval.c3.sample.tutorial1.hellomachine.HelloMachineServices;

public class HelloMachineImpl implements HelloMachine,HelloMachineServices{
    @Override
    public String salute(String name) {
        // compute the salute with the given name
        return "Hello " + name + " !";
    }
}
```

## Launch the component

Just use the component manager and provide the component to be created. Secondly, still with the component manager get the component's service and use it. Pretty simple.

```
package org.capcaval.c3.sample.tutorial1;
import org.capcaval.c3.componentmanager.ComponentManager;
import org.capcaval.c3.sample.tutorial1.hellomachine.HelloMachineServices;
import org.capcaval.c3.sample.tutorial1.hellomachine.impl.HelloMachineImpl;

public class HelloMachineMain {
    public static void main(String[] args) {

        // get the C3 component manager
        ComponentManager cm = ComponentManager.componentManager;

        // start the application, with the given component
        cm.startApplication(HelloMachineImpl.class);

        // retrieve the service of the created component
        HelloMachineServices hms = cm.getComponentService(HelloMachineServices.class);
        // use the service to retrieve a salute sentence
        String sentence = hms.salute("world");

        // display the sentence to the system output
        System.out.println(sentence);
    }
}
```