

```

package fr.istic.nplouzeau.cartaylor.api;

import java.util.Set;

public interface Configuration {

    boolean isValid();

    boolean isComplete();

    Set<PartType> getSelectedParts();

    void selectPart(PartType chosenPart);

    PartType getSelectionForCategory(Category category);

    void unselectPartType(Category categoryToClear);

    void clear();

}

package fr.istic.nplouzeau.cartaylor.api;

/**
 * @author plouzeau
 * <p>
 * A public type to organize part types in categories
 */
public interface Category {

    String getName();

}

package fr.istic.nplouzeau.cartaylor.api;

public interface PartType {

    String getName();

    Category getCategory();

}

package fr.istic.nplouzeau.cartaylor.api;

import java.util.Set;

public interface Configurator {

```

```

        Set<Category> getCategories();

        Set<PartType> getVariants(Category category);

        Configuration getConfiguration();

        CompatibilityChecker getCompatibilityChecker();
    }

    package fr.istic.nplouzeau.cartaylor.api;

    import java.util.Set;

    public interface CompatibilityChecker {

        Set<PartType> getIncompatibilities(PartType reference);

        Set<PartType> getRequirements(PartType reference);
    }

    package fr.istic.nplouzeau.cartaylor.api;

    import java.util.Set;

    public interface CompatibilityManager extends
    CompatibilityChecker {

        void addIncompatibilities(PartType reference,
        Set<PartType> target);

        void removeIncompatibility(PartType reference, PartType
        target);

        void addRequirements(PartType reference, Set<PartType>
        target);

        void removeRequirement(PartType reference, PartType
        target);
    }

```