

# JTable

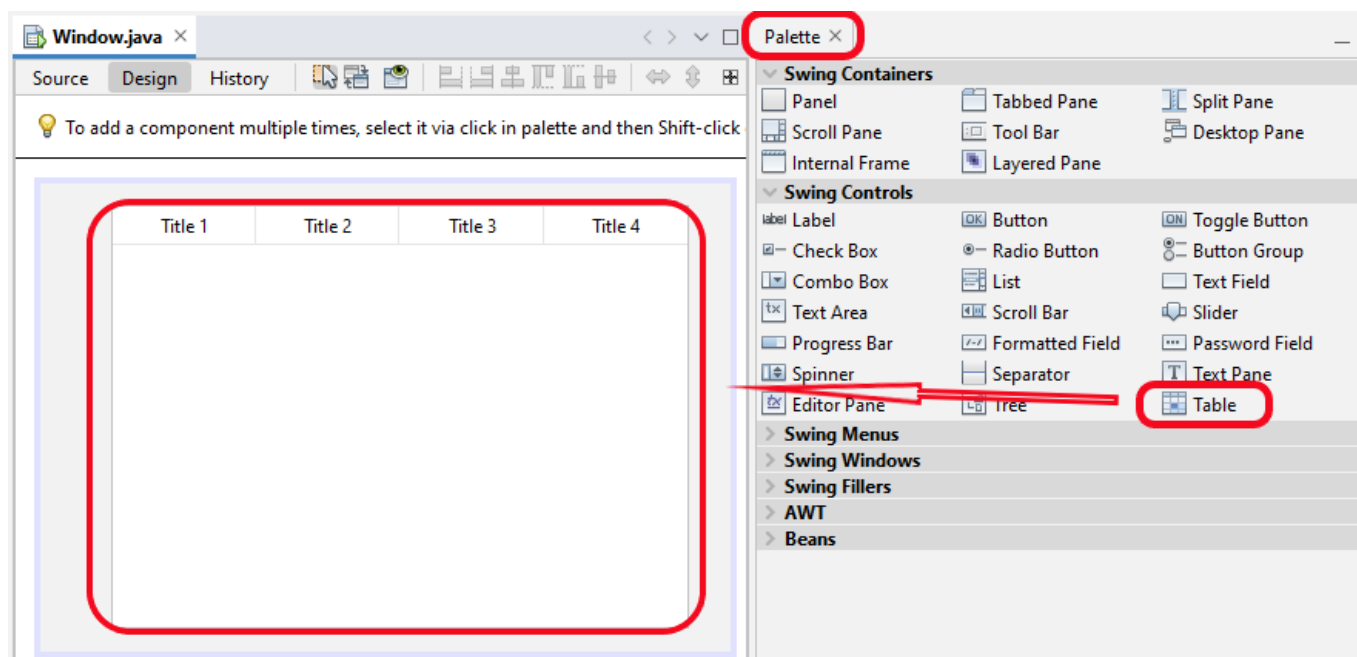
## Qu'est-ce qu'une JTable ?

Une JTable est un composant graphique qui permet de présenter un ensemble de données sous forme tabulaire.

Cette présentation à visée de **consultation** pourra donner lieu à **mise à jour, suppression, insertion** selon les souhaits du développeur. L'ensemble des valeurs présentées proviendra le plus souvent d'instances d'objets dont on obtiendra une représentation tableau.

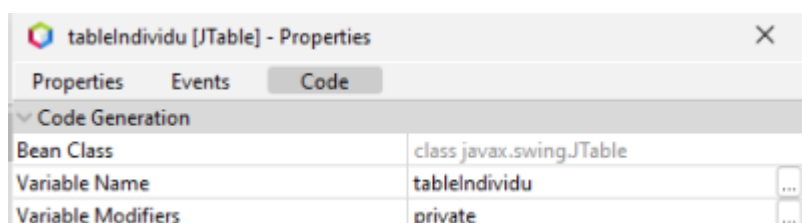
## Déposer une JTable

Dès que l'on dispose d'un container graphique, par exemple, une **JFrame**, on peut créer une JTable en mode **Design**:



## Paramétrer la JTable

Commencer par adapter le nom de la JTable dans l'onglet **code**



Une JTable est un composant graphique qui affiche des données sous forme tabulaire. En vertu des principes de structuration du code, les données à afficher dans le JTable ne sont pas portées par la JTable elle-même mais par un composant non graphique spécialisé (un modèle ... ça vous rappelle quelque chose ...?), ici un **DefaultTableModel**.

Pour alimenter la JTable en données, il faudra donc **créer un attribut de type DefaultTableModel** et le paramétrer. C'est le modèle qui fournira la structure des données en colonnes et en lignes.

```
public class MainWindow{
    private DefaultTableModel tableModel;           // attribut modèle
    ...

    public MainWindow() {
        initComponents();
        tableModel= new DefaultTableModel();        // instantiation du modèle

        String header[] = new String[] {"Colonne1", "Colonne2", "Colonne3"};
        tableModel.setColumnIdentifiers(header); // définition des colonnes en
        nombre et en intitulés
        ...
    }
    ...
}
```

## Remplir ou vider la JTable

### Les fonctionnalités essentielles

|  |  |
|--|--|
| <code>tableModel.addRow(Object[] data)</code>                            | Permet d'ajouter une nouvelle ligne dans le modèle   |
| <code>tableModel.setRowCount(0)</code>                                   | Permet de vider le modèle, si besoin   |
| <code>tableModel.setDataVector(Object[][] data, Object[] headers)</code> | Permet d'alimenter le modèle à partir d'un tableau déjà existant et un tableau des entêtes |
| <code>jtable.setModel(DefaultTableModel model)</code>                    | Associe le modèle et la JTable   |

### Exemple d'usage

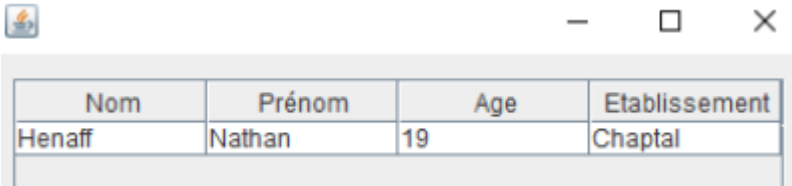
```
tableModel.setRowCount(0);
tableModel.addRow(new Object[]{"Henaff", "Nathan", 19, "Chaptal"});
jTable.setModel(tableModel);
```

### Autre exemple d'usage

```
Object [][] data = new Object [][] {
    {"Henaff", "Nathan", 19, "Chaptal"},
```

```
    {"Henriot", "Titouan", 20, "Chaptal"},  
    {"Le Gars", "Thibault", 18, "Chaptal"}  
}  
String header[] = new String[] {"Nom", "Prénom", "Age", "Etablissement"};  
tableModel.setDataVector(data, headers);  
jTable.setModel(tableModel);
```

## Exemple de résultat



The screenshot shows a standard Java Swing window with a title bar containing a minimize button, a maximize button, and a close button. The window contains a JTable with the following data:

| Nom    | Prénom | Age | Etablissement |
|--------|--------|-----|---------------|
| Henaff | Nathan | 19  | Chaptal       |

From:  
<https://wiki.siochaptalqper.fr/> - **Wiki SIO Chaptal**

Permanent link:  
<https://wiki.siochaptalqper.fr/doku.php?id=bloc2:prog:poo:jtable&rev=1699959184>

Last update: **2023/11/14 11:53**

