

Designer で データストアを使う

目次

- データストアとは何か? 2
- Designer でデータストアを使う 2
 - ストアタイプを選択する 2
 - クロスドメインリクエスト 3
 - ソースデータの場所の指定 3
 - データフィールドをマッピングする 5
 - ストアにデータをロードする 6
 - UI コンポーネントにストアをバインドする 7
- 用例 8
 - Json Store を使う 8
 - XML Store を使う 9
 - Array Sotre を使う 10

☐ データストアとは何か？

データストアは UI コンポーネントにクライアントサイドのデータキャッシュを提供します。データストアは XML ファイルのようなソースからデータを取り出し、GridPane のような UI コンポーネントと一緒に使えるようにします。

そうするためには、ストアは XML ファイルや JSON パケットなどのソースから構造化データを読み込み、UI コンポーネントからアクセスできるレコードオブジェクトの配列を生成するために DataReader を使います。読みリクエストは DataProxy によってハンドリングされます。DataProxy は、ソースへのアクセス方法や DataReader へのデータの渡し方を知っています。

データストアを設定する時、どんなフォーマットのデータが入っていて、それがどこにあるかを指定します。データソースにあるフィールドを UI コンポーネントで使えるフィールドにマップしてから、それらのフィールドを使うようにコンポーネントを設定します。

☐ Designer でデータストアを使う

データストアを使って UI コンポーネントにデータを表示するには次のようにします。

1. ソースデータのフォーマットにあったストアのタイプを選択します : Json Store, Array Store, XML Store, Direct Store
2. ストアの url 属性にどこからデータを読み込むのかを指定します。
3. データストアにフィールドを追加して、それらをソースデータにマップします。
4. データをデータストアにロードします。
5. データストアの指定したフィールドを使うように UI コンポーネントを設定します。

☐ ストアタイプを選択する

Designer では、いくつかのデータストアのタイプから選択できます。それぞれのタイプはソースからのデータを取り出し解析するのに使われる DataReader と DataProxy の種類を定義しています。

- ◆ **Json Store**—JsonReader と HTTPProxy を使って JSON パケットからデータを取り出します。
- ◆ **Arrya Store**—ArryReader と MemoryProxy を使ってローカル配列からデータ取り出します。
- ◆ **XML Store**—XmlReader と HTTPProxy を使って XML ファイルからデータを取り出します。
- ◆ **Direct Store**—JonReader と DirectProxy を使ってサーバーサイドのプロバイダからデータを取り出します。

Designer でストアを追加するには

1. Data Store タブを選択します。
2. 追加したいストアのタイプを選択します。



🚫 クロスドメインリクエスト

重要なことですが、HTTPProxy は同じドメインからのデータしか取り出せないことに注意してください。これはリモートソースからデータを取得するために Json Store や XML Store を生成できないということを意味します。クロスドメインリクエストには、ScriptTagProxy を使わなければなりません。

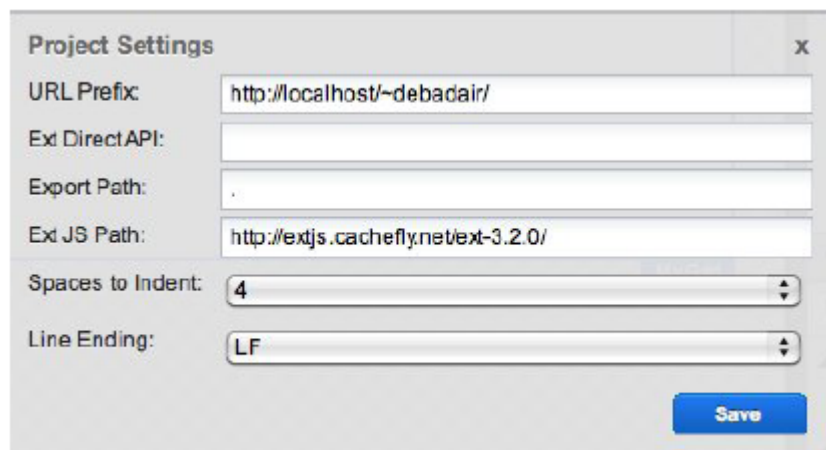
クロスドメインリクエストを手助けするために、Json パッケージにアクセスするのに ScriptTagProxy を使う JsonP Store タイプが Designer に提供されるでしょう。

📍 ソースデータの場所の指定

ストアを作成したら、ソースデータの場所を指定する必要があります。Designer では、ストアの url 属性に指定した場所はプロジェクトの設定で指定された URL prefix からの相対アドレスです。

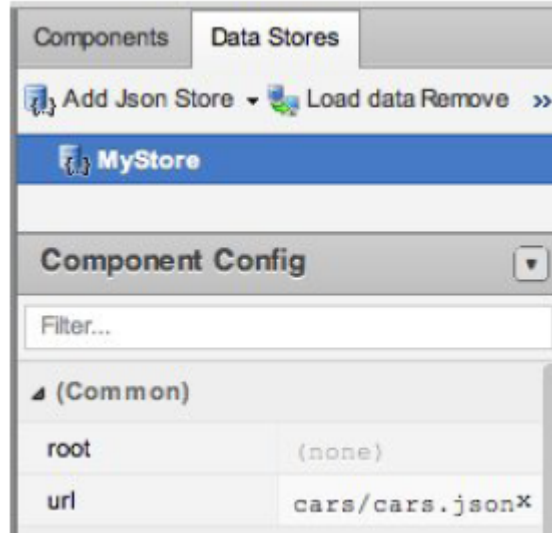
プロジェクトの URL prefix を設定するには

1. Edit メニューの Edit Preference を選択します。
2. 個々のコンポーネントに設定した url 属性の先頭に付加するべきプリフィクスを URL prefix に設定します。

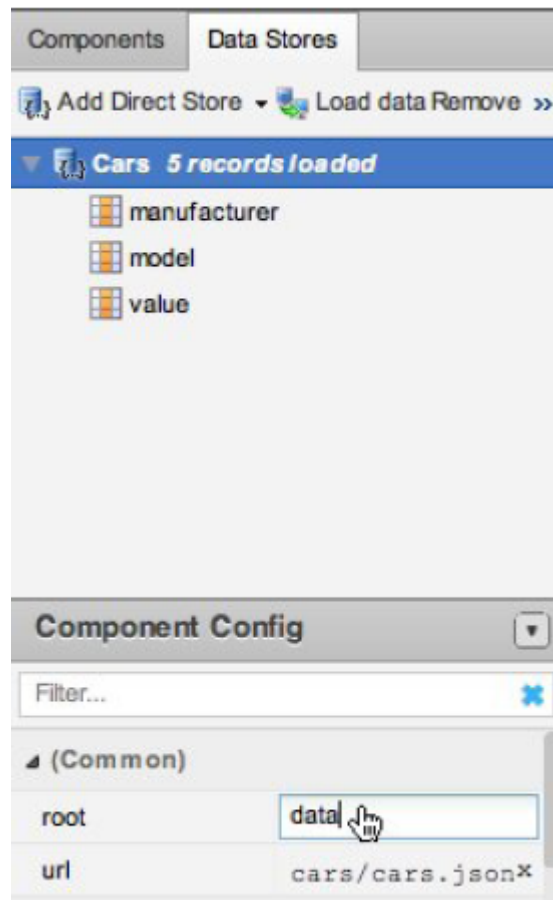


ストアのソースデータの場所を指定するには

1. Data Store タブのストアを選択します。
2. ストアの url 属性をソースデータを示すように設定します。



3. ほとんどのストアタイプでは、どこからデータを読むべきか属性の名前をリーダーに伝えるために root 属性に設定する必要があります。

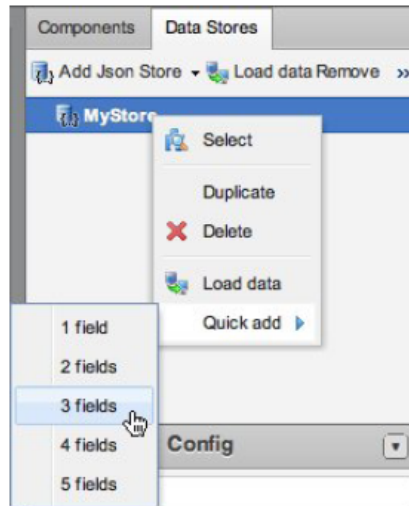


データフィールドをマッピングする

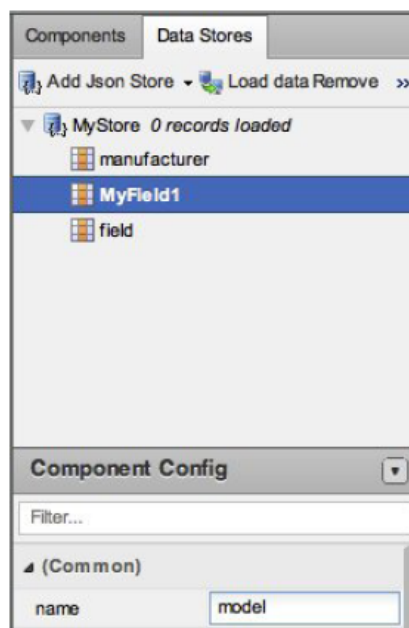
ソースからロードしたいそれぞれの元素のために、ストアにフィールドを追加する必要があります。

ストアにフィールドを追加するには

1. Data Store タブのストアを右クリックします。



2. Add Field をクリックしてストアに追加したいフィールドの数を選択します。

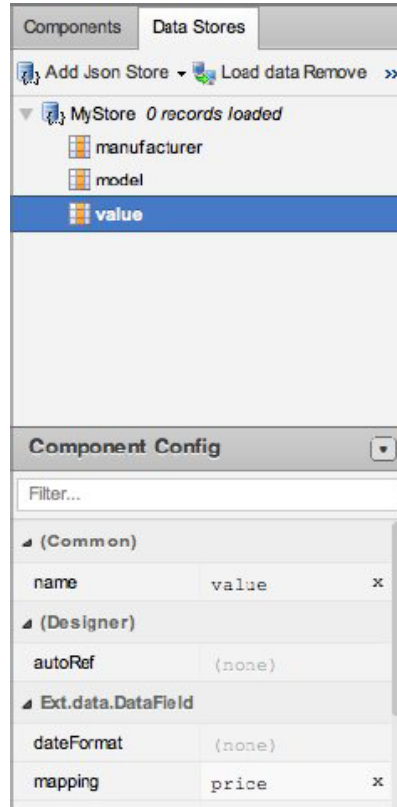


3. それぞれのフィールドの name 属性を設定します。

デフォルトでは、データソースのそれぞれのフィールドはソースデータにある同じ名前のフィールドにマップされます。しかしながら、フィールド設定の mapping 属性を指定するとどのソースフィールドでもフィールドにマップすることができます。例えば、ソースデータに登場する表記のアンダースコアを削除したいとか、それを大文字にしたいとか、フィールドに全く違う名前をマッピングするなどです。

ソースフィールドと違う名前をマッピングするには

1. Data Source タブからフィールドを選択します。
2. フィールドにマップしたいソースデータを特定するために、フィールドの mapping 属性を設定します。



また、ソースフィールドから読まれたデータが、どうフォーマットされるかも制御できます。例えば、日付フィールドの内容をどのように表示したいかを指定するには、フィールド設定で dateFormat 属性を設定します。これは PHP スタイルの日付フォーマット文字列です。詳しくは [ドキュメントの Date](#) を参照してください。

同様に、sortType を設定すると、ソートの時にそのフィールドがどう扱われるかをコントロールすることができます。あらかじめ定義された SortType 関数のどれかを指定するか、独自のソート関数を定義して使います。

ストアにデータをロードする

autoLoad 属性を enable にするとデータをストアに自動的にロードできます。これにより、生成時にストアの load メソッドが自動的に呼び出されます。

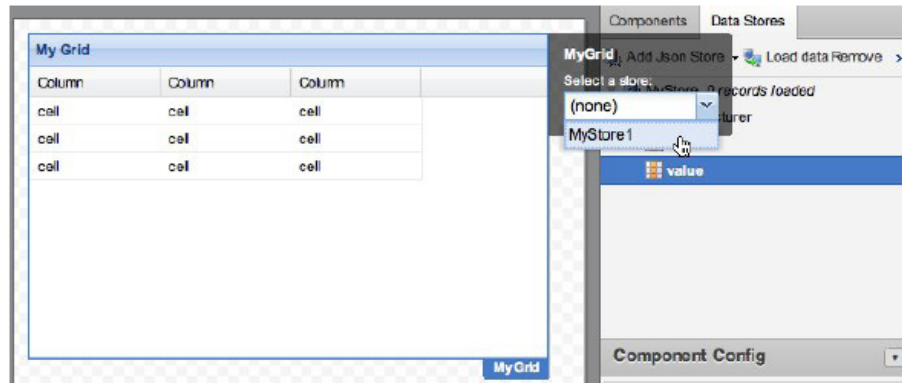
Designer では、Load Data をクリックすると強制的にソースからデータをロードできます。データがソースから読み込めなかった場合は、Designer がソースデータがあると期待していた場所を含んだエラーメッセージが表示されます。

autoLoad を enable にしない場合、アプリケーションはデータをロードするためにストアの load メソッドを明示的に呼び出す必要があります。

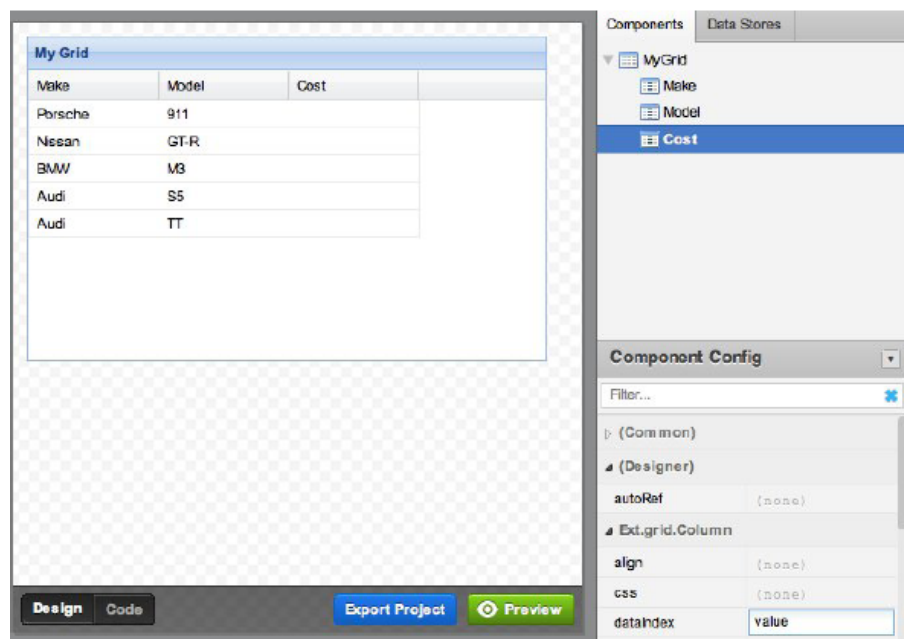
UI コンポーネントにストアをバインドする

データストアの設定ができたら、簡単にデータを表示するために UI コンポーネントにデータストアをバインドします。

1. コンポーネントのフライアウトコンフィグボタンをクリックします。
2. 使いたいデータストアを選択します。



3. コンポーネントがストアを使うように設定します。これはコンポーネントによって違います。例えば、DataGrid の場合なら、それぞれの列の dataIndex を表示したいフィールドにセットします。dataIndex を設定したらすぐにそれが表示されます。



ComboBox では、データストアの適切なフィールドに対応させるために itemId と name 属性を指定します。

データは UI コンポーネントにすぐに表示されます。表示されない場合は、

- ◆ ストアがロードできることを確かめてください。最もよくある問題はストアへのパスが正しく指定されていないことです。
- ◆ データストアの設定をチェックしてください。表示しようとしているフィールドを定義しましたか？root 属性が必要な場合それが正しく指定されていますか？

- ◆ コンポーネントの設定をチェックしてください。どのフィールドを表示するか正しく指定していますか？

注：いまのところ、接続したストアのデータは Designer のプレビューモードでは表示されません。ウェブページでどのようにデータが表示されるかを見るには、プロジェクトをエクスポートしてブラウザでアクセスしてください。

用例

次の用例は Designer でデータストアを作成し、ソースデータのさまざまな方に接続する方法を披露します。

Json Store を使う

1. ロードしたいデータを持つ Json ファイルを作成します。この例では、次のデータを持つ `customers.json` というファイルを作成します。

```
{
  customers: [
    {name: "Bradley Banks", age: 36, zip: "10573"},
    {name: "Sarah Carlson", age: 59, zip: "48075"},
    {name: "Annmarie Wright", age: 53, zip: "48226"}
  ]
}
```

2. `customers.json` ファイルをプロジェクトの URL Prefix で指定されたホストに保存します。例えば、URL Prefix が <http://localhost> に設定されていたら、<http://localhost/data/customers.json> で利用できるようにします。
3. Designer で、Data Store タブに行って **Add Json Store** を選択します。
4. ストアの root 属性に `customers` と設定します。これは、ロードしたいデータを含む Json ファイルの中で指定された配列の名前と一致します。
5. ストアの `idProperty` 属性に `name` と設定します。
6. ストアの `url` 属性にホスト上でのソースファイルの場所をプロジェクトの URL Prefix からの相対で指定します。2 でローカルホストの `data` ディレクトリに Json ストアをセーブしたので、`url` 属性には `data/customers.json` と設定します。
7. ストアコンポーネントを右クリックして、**Add Fields > 3 fields** を選択します。—それぞれ `customers` 配列の中のエレメントからアクセスしたい `name:value` のペアを設定します。
8. 最初のフィールドの `name` 属性を `name` とします。
9. 2 番目のフィールドの `name` 属性を `age` とし、その `type` 属性を `int` にします。
10. 3 番目のフィールドの `name` 属性を `zipcode` とします。これは Json ファイル中でこの値を参照するために使われる名前とは異なるので、`mapping` 属性に `zip` と設定する必要もあります。
11. ストアが選択されている状態で **Loda Data** をクリックしてください。ソースからデータのロードに成功したら、Data Stores タブに何件のレコードがロードされたか

ステータスメッセージが表示されます。データがロードできなかった場合は、Designer がデータをロードしようとした URL を含んだエラーメッセージが表示されます。

これでストアを UI コンポーネントにバインドして、ストアのフィールドを使って動的にコンポーネントにデータをロードすることができます。

XML Store を使う

1. ロードしたいデータを入れた XML ファイルを作成します。この例では、次のデータを入れた product.xml というファイルを作ります

```
<?xml version="1.0" encoding="UTF-8"?>
<products xmlns="http://example.org">
  <product>
    <name>Widget</Name>
    <price>11.95</Price>
    <imagedata>
      <url>widget.png</Url>
      <width>300</Width>
      <height>400</Height>
    </imagedata>
  </product>
  <product>
    <name>Sprocket</Name>
    <price>5.95</Price>
    <imagedata>
      <url>sc.png</Url>
      <width>300</Width>
      <height>400</Height>
    </imagedata>
  </product>
  <product>
    <name>Gadget</Name>
    <price>19.95</Price>
    <imagedata>
      <url>widget.png</Url>
      <width>300</Width>
      <height>400</Height>
    </imagedata>
  </product>
</products>
```

2. products.xml ファイルをプロジェクトの URL Prefix で指定されたホストに保存します。URL Prefix が <http://localhost> と設定されていたら、<http://localhost/data/products.xml> から利用できるようにします。

3. Designer で Data Stores タブに行き **Add XmlStore** を選択します。
4. ストアの url 属性にホスト上でのソースファイルの場所をプロジェクトの URL Prefix からの相対で指定します。2 でローカルホストの data ディレクトリに XML ファイルをセーブしたので、url 属性には data/products.xml と設定します。
5. ストアの record 属性にロードするデータを含んだ XML エレメントの名前 (Product) を設定します。
6. ストアコンポーネントを右クリックして、**Add Fields > 3 fields** を選択します。— それぞれの Product の中のアクセスしたいそれぞれのサブエレメントを設定します。
7. 最初のフィールドの name 属性を name とし、その mapping 属性を Name とします。mapping は大文字小文字を区別し、エレメント名に一致しなければならないことに注意してください。
8. 2 番目のフィールドの name 属性を price とし、その mapping 属性を Price に、その type 属性を float にします。
9. 3 番目のフィールドの name 属性を imageUrl とし、mapping 属性を ImageData > Url とします。ImageData の下位要素の Url にアクセスするのに DomQuery セレクタを使用することに注意してください。
10. ストアが選択されている状態で **Loda Data** をクリックしてください。ソースからデータのロードに成功したら、Data Stores タブに何件のレコードがロードされたかステータスメッセージが表示されます。データがロードできなかった場合は、Designer がデータをロードしようとした URL を含んだエラーメッセージが表示されます。

これでストアを UI コンポーネントにバインドして、ストアのフィールドを使って動的にコンポーネントにデータをロードすることができます。

Array Sotre を使う

1. ロードしたいデータを持つ Json ファイルを作成します。この例では、次のデータを持つ contacts.json というファイルを作成します。

```
[
  ["Ace Supplies", "Emma Knauer", "555-3529"],
  ["Best Goods", "Joseph Kahn", "555-8797"],
  ["First Choice", "Matthew Willbanks", "555-4954"]
]
```

2. contacts.json ファイルをプロジェクトの URL Prefix で指定されたホストに保存します。URL Prefix が <http://localhost> と設定されていたら、<http://localhost/data/contact.json> から利用できるようにします。
3. Designer で Data Stores タブに行き **Add ArrayStore** を選択します。
4. ストアの idIndex 属性を 0 にします。これは各行の最初のエレメント (contact name) のインデックスとして使われます。
5. ストアの url 属性にホスト上でのソースファイルの場所をプロジェクトの URL Prefix からの相対で指定します。2 でローカルホストの data ディレクトリに Json ファイルをセーブしたので、url 属性には data/contacts.json と設定します。

6. ストアコンポーネントを右クリックして、**Add Fields > 3 fields** を選択します。—ソースファイルの中の行の配列から到達したいエレメントを設定します。
7. 最初のフィールドの name 属性を name とします。
8. 2 番目のフィールドの name 属性を contact とします。
9. 3 番目のフィールドの name 属性を phone とします。
10. ストアが選択されている状態で **Loda Data** をクリックしてください。ソースからデータのロードに成功したら、Data Stores タブに何件のレコードがロードされたかステータスメッセージが表示されます。データがロードできなかった場合は、Designer がデータをロードしようとした URL を含んだエラーメッセージが表示されます。

これでストアを UI コンポーネントにバインドして、ストアのフィールドを使って動的にコンポーネントにデータをロードすることができます。