

Designer で コンポーネントを使う

目次

• Toolbox には何がある？	2
• コンテナ	2
• フォームフィールド	5
• Grid	8
• Menu	11
• Standard	11
• Toolbar	12
• Tree	14
• Views	15
• フォームの構築	17
• 簡単なフォームを構築する	17
• ラジオボタンやチェックボックスのグループを追加する	19
• 段組でフィールドを配置する	21
• フィールドを水平に並べる	22
• ComboBox を配置する	23
• メニューを構築する	25
• メニューバーの構築	25
• サブメニューを作る	28
• ツリーを配置する	29

☰ Toolbox には何がある？

Designer は全ての標準 ExtJS UI コンポーネントをサポートします。将来のリリースでは、カスタムコンポーネントもツールボックスに追加できるようになるでしょう。このセクションでは、標準のツールボックスコンポーネントの概要を解説します。それぞれのコンポーネントに関する詳細は、ExtJS API ドキュメントをご覧ください。

☰ コンテナ

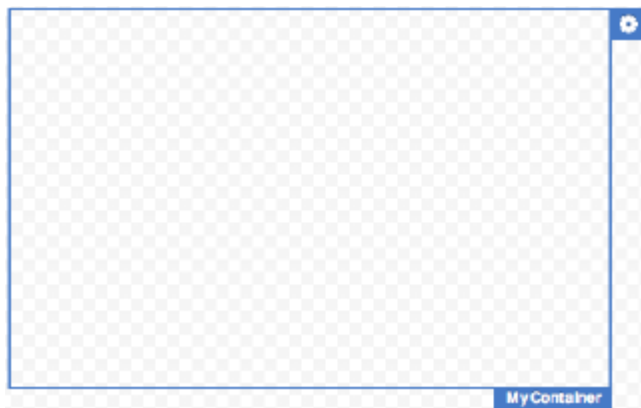
UI を構築するには、コンテナから始めてディスプレイやコントロールをそこに追加していきます。ExtJS は Window や Panel や Viewport などの様々な UI コンテナを提供します。コンテナの layout オプションを設定すると、コンテナの中でどのようにコンポーネントが配置されるかをコントロールできます。さまざまなレイアウトオプションがどのように動作するのかをみるには、[ExtJS レイアウトブラウザ](#)をよく調べてください。また“Getting Started with Designer”にもレイアウトの紹介があります。

Tip : コンテナをネストしている場合、階層の中に余分なコンテナを追加していないか確認してください。例えば Border レイアウトを使っているビューポートの中に TreePanel や GridPanel を表示したい場合、それらを直接ビューポートの中に追加してかまいません。左とセンターにパネルを追加してその中に TreePanel や GridPanel を入れる必要はありません。

Designer で UI を構築し始めるには、コンテナをキャンバスにドラッグして、コンポーネントをコンテナに追加します。Designer は不正なコンポーネントをコンテナに追加することを防ぎます。例えば Viewport や Window はトップレベルのコンポーネントでなければならず、他のコンテナにネストすることはできません。

コンテナをキャンバスの空いている場所にドラッグすると新しいトップレベルのコンポーネントを作成します。プロジェクトをエクスポートする際には、トップレベルのコンポーネントごとに分離したクラスファイルが自動的に生成されます。また、Promote to Class オプションを使うと、ネストしたコンポーネントを分離したクラスとしてエクスポートできます。これにより、全てのコードを一つのファイルに持たせるかわりに、複雑なインターフェースをいくつかのより小さな実装ファイルに分割して生成し保守することができます。このことで簡単にカスタムコンポーネントを再利用できます。

☰ Container



Container は、他のコンポーネントを配置できる最もシンプルなコンポーネントです。他のすべてのコンテナタイプは Container クラスを拡張したものです。Container はシンプルな論理コンテナです。Panel や Window と違って、なんの特徴もありません。たいていはより専門化した

コンテナを使うことになりましたが、追加機能の必要がない（あるいは欲しくない）場合には軽快に使用できます。例えば、フォームの中で段組レイアウトを作成するとき Container が使えます。

Container のデフォルトレイアウトは Container レイアウトで、ネストされたコンポーネントを単純にそのまま描画します。Container のデフォルトレイアウトを使った場合は、ネストされたコンポーネントはコンテナがリサイズされたときにリサイズされません。

FieldSet

FieldSet は FormPanel の関連するフィールドをグループ化するために使われます。FieldSet の title 属性を設定すると FieldSet のフレームにヘッダーとして表示されます。

普通はフォームフィールドを FieldSet に配置しますが、ネストしたコンテナを配置

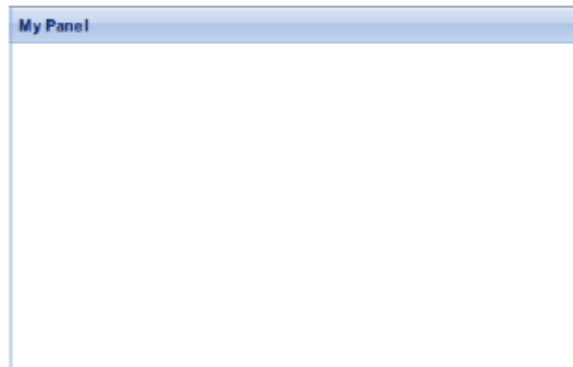
することもできます。例えば、マルチコラムレイアウトを作成する時にはネストした Container コンポーネントを FieldSet の中で使います。フィールドを追加するネストした Container も Form レイアウトを使う必要があることに注意してください。

FormPanel

FormPanel はフィールドとラベルをレンダリングするための Form レイアウトを使った、専用コンテナです。FormPanel には各種のフォームフィールドの他に、Container や FieldSet といったコンテナを追加できます。例えば、マルチコラムのフォームを構成するためにネストした Container を使います。ネストしたコンテナは同様に Form レイアウトでなければなりません。

FormPanel は内部的にファイルアップロードのハンドリングや、データバリデーションやサブミッションのために BasicForm を使います。

Panel



Panel はアプリケーション指向のユーザインタフェース向けの基本的な構成要素です。Panel はどんなタイプのコンテナにも入れることができ、分離したヘッダー、フッター、ボディーセクションを持たせることができます。

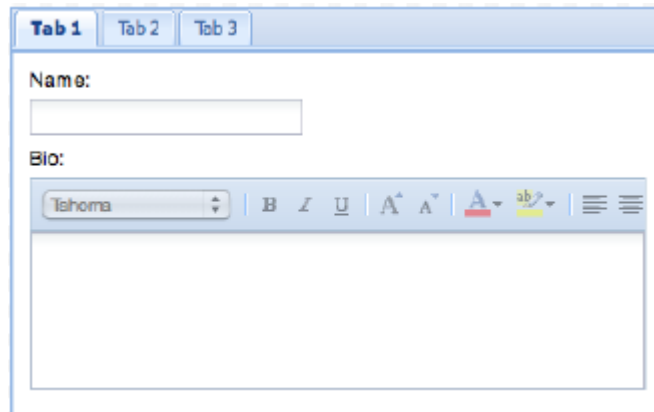
UI に直接 Panels を加えることができますが、FormPanel, TabPanel, GridPanel, TreePanel と

いった多くの専門化しているタイプのパネルが利用可能です。また Window と FieldSet コンテナも Panel を拡張したものです。

デフォルトで Panel は Container レイアウトを使います。ネストされたコンポーネントは Panel クラスに配置された順番で単純にレンダリングされます。

ネストされたコンポーネントの位置やサイズをコントロールするには、適切なレイアウトを選択します。

TabPanel



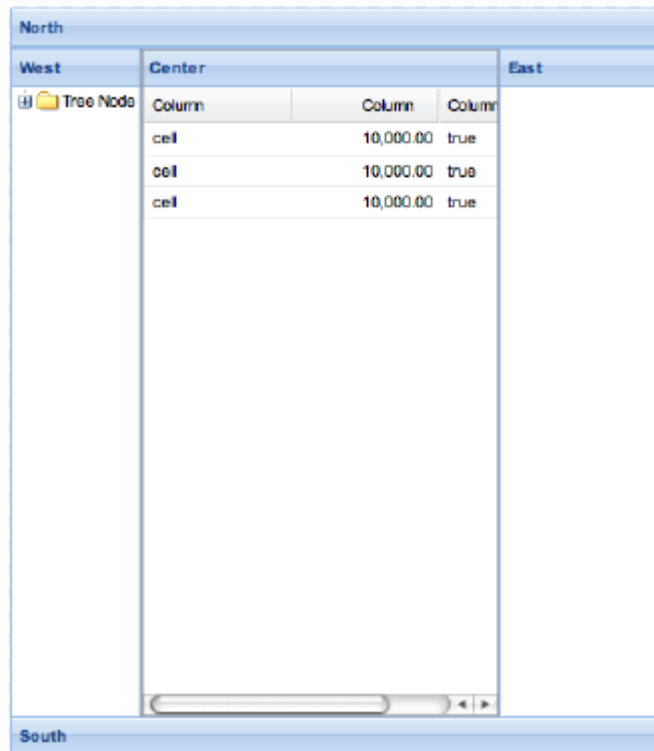
TabPanel は配置されたコンポーネントを分離したタブに表示する Card レイアウトを使ったパネルです。TabPanel の title 属性は表示されません。

注： TabPanel はヘッダーとフッタースペースがタブ選択ボタンに使われます。ヘッダーを表示する必要がある場合は、Fit レイアウトの Panel コンテナに

TabPanel をラップします。

Designer では、キャンバスに TabPanel を追加した時、デフォルトで3つのタブコンポーネントがあります。TabPanel の上にコンポーネントをドラッグすることで、サブコンポーネントを追加することができます。

☰ Viewport

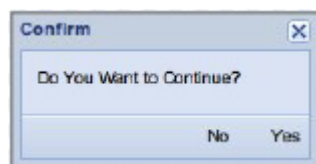


Viewport はブラウザウィンドウのアプリケーションエリア全体を描画するために使います。Viewport は自動的に自分自身をブラウザビューポートのサイズにします。1 ページあたり 1 つの Viewport しか持つことができませんが、レイアウトが設定され、コンポーネントが配置されたパネルを配置することができます。Viewport はスクロールできません。—スクロールが必要な場合は Viewport の中にスクロール可能な子コンポーネントを配置して使います。

通常 ViewPort には Border レイアウトを割り当て、パ

ネルの region 属性に North, South, West, East, Center をセットして位置指定をします。region が指定されない場合のデフォルトは Center です。

☰ Window



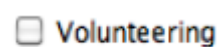
Window はリサイズができてドラッグできるパネルです。Window はまたビューポートいっぱいに最大化したり、最小化したり、元のサイズに戻すことができます。普通のパネルと違って Window は移動したり閉じたりできます。Window は主にダイアログやエラーを表示するのに使われます。

ます。

☰ フォームフィールド

フォームを構築するために、フォームフィールドコンポーネントを FormPanel に追加します。FieldSet を使って関連するフィールドをグループ化することもできます。マルチコラムのフォームを作るためにコラムに Container を使うこともできます。フォームのデザインに関して詳しくは、「フォームを構築する」の章をご覧ください。

☰ Checkbox



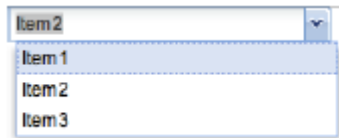
Checkbox は一つのチェックボックスフィールドを表示します。チェックボックスのラベルを設定するには boxLabel 属性をセット

します。

複数のチェックボックスを Container に追加してチェックボックスグループを作ることができます。

注：ExtJS 3.x の RadioGroup および CheckboxGroup に関する問題により、これらのコンポーネントは Designer の Toolbox に含まれていません。Designer においてラジオボタンやチェックボックスのグループを作るときのコンテナの使い方については「ラジオボタンやチェックボックスのグループを追加する」の項をご覧ください。

☰ ComboBox



ComboBox はユーザーがリストからアイテムを選択できるようにします。

ComboBox のアイテムを設定するには、データストアに接続させます。詳しくは、「ComboBox を配置する」の項をご覧ください。

ComboBox の高さは常に自動的にセットされます。ComboBox は幅のみ指定可能です。ですので、

1. anchor, form, fit レイアウトで使わない。
2. EditorGrid コラムの中に置かない。

☰ CompositeField



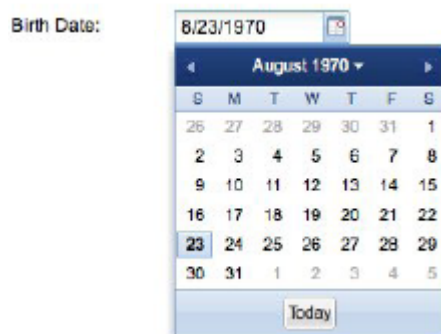
CompositeField はフォームの同じ行に複数のフィールドを簡単に表示できます。

よく使われるのは複数部分の名前フィールドです。ですが、CompositeField は TextField だけではなく、あらゆるタイプのフォームフィールドを配置することができます。

fieldLabel が指定されていない場合は、デフォルトで子コンポーネントの fieldLabel 属性のリストがラベルになります。(すべてのフィールドと同様、FormPanel に配置されたときに CompositeField のラベルも自動的にレンダリングされます)

Designer では、CompositeField をキャンバスにドラッグしたら、デフォルトで 1 つの TextField がセットされています。CompositeField にフィールドをドラッグしたり、既存のフィールドを複製したりしてフィールドを追加できます。

☰ DateField



DateField は ExtJS アプリケーションにデロピッカーを提供します。ユーザーが手動で入力した日付の自動データバリデーションも提供します。

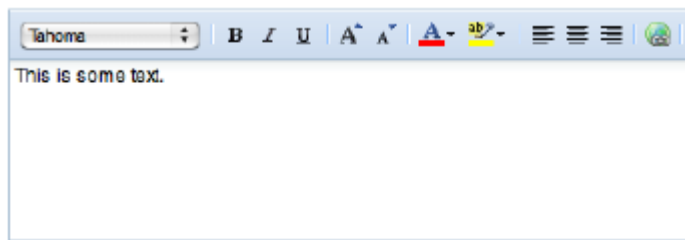
DisplayField

DisplayField は表示のみのテキストをレンダリングします。フォームでは値のチェックも送信もされません。

Hidden

Hidden は、フォームに表示されないフィールドです。フォームが送信されたときにデータを送るために使われます。Designer で Hidden フィールドをキャンバスにドラッグしても何も表示されませんが、Component タブには表示されます。

HtmlEditor



HtmlEditor は、軽量な WYSIWYG な HTML エディタで、ユーザーに書式付きのテキストを送信させることができます。エディタツールバーにツールチップが

定義されていますが、それを有効にするには QuickTips を初期化する必要があります。

NumberField

Quantity:

NumberField は、数値データの値のチェックを自動的に実行でき、数値しか入力できないようになっているテキストフィールドです。

属性を設定して小数点や負の数を許可するかどうかをコントロールしたり、最大値を設定したりできます。小数値を許可する場合は、精度や区切り文字を設定できます。

Radio

Yes
 No

Radio は一つのラジオボタンを表示します。ボタンのラベルを設定するには boxLabel 属性を設定します。ラジオボタングループを作るには、複数の Radio コンポーネントを Container に追加します。

グループの中でラジオボタンが一つしか選べないようにするには、それぞれのボタンの name 属性に同じ値を設定します。

注：ExtJS 3.x の RadioGroup および CheckboxGroup に関する問題により、これらのコンポーネントは Designer の Toolbox に含まれていません。Designer においてラジオボタンやチェックボックスのグループを作るときのコンテナの使い方については「ラジオボタンやチェックボックスのグループを追加する」の項をご覧ください。

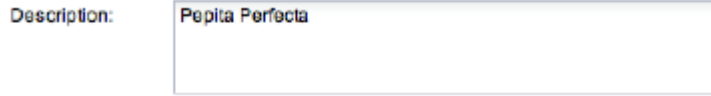
SliderField

Level: 

SliderField はスライダーコントロールが使えるようにします。数値データを入力させる時に NumberField の代わりになります。

Designer で SliderField を追加した時、選択された値を表示する useTips が有効に設定され、最小値は 0 で最大値は 100 で増分は 1 に設定されています。

☰ TextArea



TextArea は大量のテキストを入力できるように複数行のテキストが入力をサポートするテキストフィールドです。

書式付きのテキストを入力させたい場合は、HtmlEditor コンポーネントを代わりに使ってください。

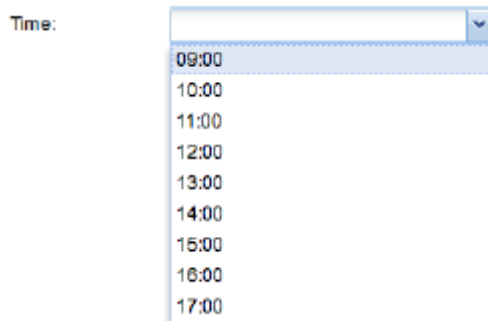
☰ TextField



TextField はベーシックなテキスト入力フィールドです。

TextField 自身がよく使われるフォームエレメントですが、それに加えていくつかの特化したフィールドタイプ (NumberField, TextArea, TriggerField, ComboBox など) の構成部品としても使われています。TextField は値のチェックを標準でサポートします。値チェック動作のカスタマイズに関する詳細は、ExtJS API ドキュメントをご覧ください。

☰ TimeField



TimeField は時間指定に特化した ComboBox です。minValue と maxValue を設定すると時間範囲を設定できます。デフォルトではリストには 15 分間隔で表示されます。increment 属性を設定すると間隔を設定できます。

TimeField は Date.js で定義された構文解析とフォーマット構文で指定された時間と日付のフォーマットをサポートします。入力

が期待したフォーマットに合致していない場合、TimeField は自動的に他のフォーマットを使って解析しようとします。デフォルトではフォーマットは "g:i A" です。これは 12 時間表示 (例: 3:15PM) で時間を表示します。24 時間表示を使う場合は、format 属性に "H:i" を設定します。

☰ TriggerField

TriggerField は TextField をラップして ComboBox についているようなクリック可能なトリガーボタンを追加したものです。

デフォルトで TriggerField は ComboBox とよく似ていますが、トリガーには動作が割り当てられていません。onTriggerclick イベントをオーバーライドするか、TriggerField を拡張してカスタムコンポーネントを実装するかしてカスタムアクションを提供します。ComboBox は標準のコンボボックスの挙動を提供するように TriggerField を拡張したものです。TimeField も特化した TriggerField です。

☰ Grid

ストアのデータをインタラクティブなテーブルコンポーネントに表示するときに GridPanel を使います。GridPanel は標準でリサイズ、項目のソート、項目のドラッグ&ドロップをサ

ポートします。水平スクロール、単一選択および複数選択をサポートします。EditorGridPanel はさらにインライン編集をサポートします。

GridPanel が提供する項目の管理機能のすべてを必要とするわけではないけれど、データのフォーマットやレイアウトを十分にコントロールしたいときには、グリッドの代わりに ListView を使用した方がいいでしょう。

BooleanColumn

BooleanColumn は、論理値データをグリッドに表示する特化したコラムです。

CellSelectionModel

CellSelectionModel は、列全体を選択するのではなく GridPanel のそれぞれのセルを選択するときに使うセレクションモデルです。

CheckboxSelectionModel

CheckboxSelectionModel は、Grid に行を選択／非選択できるようなチェックボックスのコラムを追加した RowSelectionModel です。グループの中の洗濯したアイテムを移動したり削除したりするような動作を可能にするために使われます。

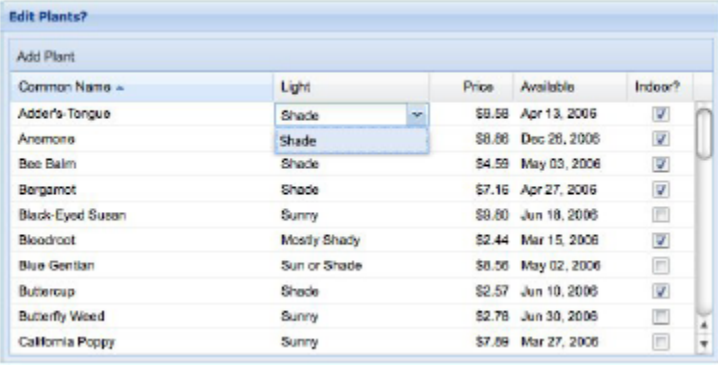
Column

Column は GridPanel のコラムがどのように表示されるかをコントロールします。Column は生のデータを表示できます、また ExtJS は論理値、数値、日付、テンプレート化されたデータを表示させる特化したタイプの Column を提供します。

DateColumn

DateColumn は、グリッドに日付を表示することに特化した Column です。format 属性を指定して日付のフォーマットをコントロールします。DateColumn は Date.js で定義された構文解析とフォーマット構文をサポートします。

EditorGridPanel



Common Name	Light	Price	Available	Indoor?
Adonis-Tongue	Shade	\$8.58	Apr 13, 2006	<input checked="" type="checkbox"/>
Anemone	Shade	\$8.88	Dec 26, 2006	<input checked="" type="checkbox"/>
Bee Balm	Shade	\$4.59	May 03, 2006	<input checked="" type="checkbox"/>
Bergamot	Shade	\$7.16	Apr 27, 2006	<input checked="" type="checkbox"/>
Black-Eyed Susan	Sunny	\$8.80	Jun 18, 2006	<input type="checkbox"/>
Bloodroot	Mostly Shady	\$2.44	Mar 15, 2006	<input checked="" type="checkbox"/>
Blue Gentian	Sun or Shade	\$8.56	May 02, 2006	<input type="checkbox"/>
Buttercup	Shade	\$2.57	Jun 10, 2006	<input checked="" type="checkbox"/>
Butterfly Weed	Sunny	\$2.78	Jun 30, 2006	<input type="checkbox"/>
California Poppy	Sunny	\$7.89	Mar 27, 2006	<input type="checkbox"/>

EditorGridPanel は、セルを編集可能にした GridPanel です。GridPanel の Column が編集できるためには、適切なエディタ（単にデータを編集するのに使用されるフォームフィールドです）が構成されている必要があります。

GridPanel

Company	Price	Change	% Change	Last Updated
3m Co	\$71.72	0.02	0.03%	09/01/2010
Alcoa Inc	\$29.01	0.42	1.47%	09/01/2010
Altria Group Inc	\$63.01	0.28	0.34%	09/01/2010
American Express Company	\$62.55	0.01	0.02%	09/01/2010
American International Group, Inc.	\$64.13	0.31	0.49%	09/01/2010
AT&T Inc.	\$31.01	-0.48	-1.54%	09/01/2010
Boeing Co.	\$75.43	0.53	0.71%	09/01/2010
Caterpillar Inc.	\$67.27	0.92	1.39%	09/01/2010
Crigroup, Inc.	\$46.37	0.02	0.04%	09/01/2010
E. I. du Pont de Nemours and Company	\$40.48	0.51	1.28%	09/01/2010
Exxon Mobil Corp	\$68.10	0.43	0.64%	09/01/2010
General Electric Company	\$34.14	-0.08	-0.23%	09/01/2010
General Motors Corporation	\$30.27	1.09	3.74%	09/01/2010
Hewlett-Packard Co.	\$30.53	-0.03	-0.08%	09/01/2010

GridPanel は行と列の表形式データを表示します。GridPanel が表示するデータはストアから読み込まれます。GridPanel の ColumnModel はグリッドの列へのデータのレンダリング方法をコントロールします。そして Column コンポーネントは個々の列をレンダリングするのに必要な設定情報をカプセル化します。

GridView

GridView は、GridPanel のユーザーインターフェースをカプセル化したものです。GridView を GridPanel に追加すると、グリッドのユーザーインターフェース要素にアクセスでき、特別な表示効果が可能になります。

GroupingView

GroupingView はグリッドの関連する行をグループ化してグループを展開したり折りたたんだりできる GridView です。例えば、顧客名と住所が表示されているグリッドがあって、市や州でグループ化したいような場合に使います。

グループ化を可能にするには、GroupingStore を使うように設定します。GroupingStore はデータに加えてグループフィールドとディレクションを指定するストアです。

NumericColumn

NumericColumn は、グリッドに数値を表示する Column です。format 属性を指定して数値のフォーマットをコントロールできます。フォーマット文字列は Ext.util.Format.number に準じて定義されます。

RowSelectionModel

RowSelectionModel は GridPanel で使われるデフォルトのセレクションモデルです。複数選択、キーボード選択、ナビゲーションをサポートします。複数選択を無効にするには singleSelect 属性を有効にします。moveEditorOnEnter 属性を無効にすると、Enter と Shift+Enter で行移動しないようにします。CellSelectionModel や CheckboxSelectionModel も参照してください。

TemplateColumn

TemplateColumn は、指定されたテンプレートでレコードのデータを処理してグリッドに表示する値を生成する Column です。テンプレートは tpl 属性にセットします。

Menu



Designer でメニューを構築するには、Menu コンポーネントをボタンに追加します。メニューバーを構築するには、ツールバーにそれぞれのメニューのボタンを追加します。そしてそれぞれのボタンの Menu コンポーネントを設定します。メニューの構築について詳しくは「メニューを構築する」の章を参照してください。

CheckItem

CheckItem はチェックボックスまたはラジオボタンでメニューアイテムのオン/オフをトグルできる MenuItem です。group 属性がセットされていると、同じグループ名の全てのアイテムはシングルセレクトのラジオボタンと同様に取扱われます。

Menu

Menu はメニューアイテムのコンテナです。メニューを作成するには Menu を Button コンポーネントに追加し、CheckItem、MenuItem、Separator、TextItem コンポーネントを Menu に追加します。

注：Menu ではメニューアイテムのコンポーネントを使うのが普通ですが、Menu にはどんなタイプのコンポーネントを追加することもできます。

Menu Item

メニューアイテム (Item) は標準の選択できる Menu オプションです。

Separator

Separator は Menu に追加できる区切りです。メニューアイテムの論理的なグループを分けるのに使います。

TextItem

TextItem は Menu に追加できる選択できない文字列です。TextItem はメニューアイテムのヘッダーとして使われます。

Standard

これらのコンポーネントは UI の基本的構成要素を提供します。

BoxComponent

BoxComponent は幅と高さまたはレイアウトでの管理を使ってサイズ指定される全てのコンポーネントのベースです。BoxComponent は直接使われることはありません、普通は Container とか特化した BoxComponent (Button や Field など) を使います。

Button

Button はアイコン、文字列、またはその両方を表示できます。Button の icon 属性を設定するとボタンにイメージを指定できます。アイコンと文字列を表示させるには、iconCls 属性にラベルの CSS クラス x-btn-text-icon を設定する必要もあります。

Buha メニューを構築する際にも使います。メニューはネストした Menu コンポーネントを持つ単なる Button コンポーネントです。メニューバーを作るには、それぞれのメニューに対応する Button をツールバーに追加して、それぞれのボタンの Menu コンポーネントを設定します。

Designer では、Button のフライアウトコンフィグボタンをクリックするか、手動で Menu コンポーネントをボタンの上にドラッグしてメニューの設定ができます。

☰ CycleButton

CycleButton は CheckItems のグループを持つ SplitButton です。CycleButton はクリックするとアイテムを順次切り替えます。プルダウンしてあるアイテムをクリックすると、ボタンのテキストはクリックされたアイテムのテキストに替わります。

☰ Label

Label コンポーネントにはフォームのコンポーネントを表す文字が設定されます。普通 Label を直接使うことはありません。Component の Label は自動的に生成され fieldLabel 属性を通して設定されます。(Checkbox フィールドでは、チェックボックスの横に表示される文字を boxLabel 属性で指定します) Component のラベルは Form レイアウトを使ったコンテナに描画される時だけ表示されます。

☰ MultiSlider

MultiSlider は複数のつまみをサポートしたスライダーコントロールです。MultiSlider はどのコンテナにも配置でき、縦／横いずれでも配置できます。複数のつまみのスライダーを作るには、単一の値を指定するのではなく values 属性に配列を指定します。スライダーをフォームで使うときは普通は SliderField を使います。

☰ ProgressBar

ProgressBar はタスクの進行状況を表示します。ProgressBar は手動と自動の 2 つのモードをサポートします。手動モードはタスクの進行状況を明示的に更新できます。予測できるマイルストーンがある操作の全体の進行状況を表示する時には手動モードを使います。自動モードでは一定時間がすぎるか、クリアされるまで単に走る。一定時間か非同期処理で進行状況を表示したい場合に自動モードを使います。

☰ Spacer

Spacer はレイアウト上に固定の空白スペースを追加する方法を提供する BoxComponent です。

☰ SplitButton

SplitButton はボタンのクリックイベントとは別にイベントを発生できるドロップダウン矢印のあるボタンです。CycleButton は SplitButton をメニューアイテムのセットをサイクルできるように拡張したものです。ComboBox は標準のコンボボックスの動作を実装するために SplitButton を使っています。



Toolbar

ボタンやメニューや他のコントロールの集まりを表示するツールバーを作るには、Toolbar コンポーネントを追加して Toolbar にコントロールコンポーネントを追加します。Toolbar は普通は Panel コンテナに追加します。上部 (tbar)、下部 (bbar)、フッター (fbar) にドッキングできます。

☰ ButtonGroup

ButtonGroup はボタン用の Panel コンテナです。概念的には ButtonGroup はユーザーが選択する関連項目のコレクションを持っているという点においてメニューに似ています。あ

らゆるタイプのボタン (Button, CycleButton, SplitButton) を ButtonGroup に追加できます。しかしながら Menu と違って他のコントロールコンポーネントを追加できません。

☰ Fill

Fill は、いくつかのコンポーネントを左寄せにし、残りを右寄せにしたいときに Toolbar に追加します。Toolbar コンポーネントの間に Fill を挿入したら、Fill に続く全てのコンポーネントは右寄せになります。例えば、キャンセルボタンと送信ボタンを buttonAlign 属性が left にセットされているツールバーに追加します、その間に Fill コンポーネントを追加してキャンセルボタンを左寄せに送信ボタンを右寄せにすることができます。

The screenshot shows a form titled "My Form" with the following elements:

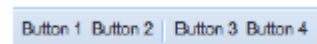
- Input fields for "First, Last:" and "Email:".
- A section titled "Do You Want to Receive Our Newsletter?" with two radio buttons: "Yes, I want to know what's going on!" (selected) and "No Thanks."
- A section titled "What Are You Interested In?" with six checkboxes: "Lessons", "Camps", "Clinics", "Private Events", "Volunteering", and "Sponsoring".
- A dropdown menu for "How Did you Hear About Us?".
- "Cancel" and "Submit" buttons at the bottom.

☰ PagingToolbar



PagingToolbar は、大きなデータセットをページングをコントロールするツールバーです。PagingToolbar は、ソースからデータストアへデータのブロックをロードすることによって、自動ページングをコントロールします。一度に表示したいアイテムの数を pageSize 属性にセットします。PagingToolbar は GirdPanel と一緒に使うようにデザインされています。

☰ Separator

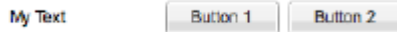


Separator は Toolbar のコンポーネントの間に区切りを挿入するアイテムです。Separator は Toolbar 上のボタンを論理的なグループに分けるのに使います。ButtonGroup も同様にボタンをグループ化するのに使えます。

☰ Spacer

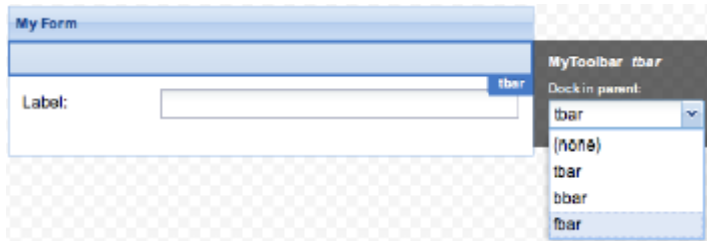
Spacer は Toolbar のコンポーネントの間に任意大きさのスペースを挿入するアイテムです。スペースの大きさをコントロールするには、Spacer の width 属性を設定します。もしも左寄せの Toolbar の一部のコンポーネントを右寄せしたい場合は Spacer ではなく Fill を挿入します。

Text



Text は選択不可能な文字列をラベルやヘッダとして Toolbar に挿入するアイテムです。

Toolbar

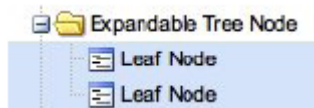


Toolbar はボタンやメニューなどのコントロールコンポーネントのコンテナです。Toolbar はパネルの上 (tbar) , 下 (bbar) , フッタ (fbar) にドッキングできます。Toolbar は、パネルの上部にメニューバーを表示したり、ダイアログウィンドウの下にボタンを表示したりする時に使います。

Tree

折りたたみ可能なリストに階層的なデータを表示するときに TreePanel コンテナを使います。TreePanel コンテナはルートノードといくつもの子ノードを持つことができます。静的なツリーも構築できますし、TreeLoader を使ってツリーに非同期的にデータをロードすることもできます。

AsyncTreeNode



AsyncTreeNode は TreeLoader を使った非同期的なデータローディングをサポートする TreeNode です。ノードが展開されたときに、データが TreeLoader に指定された URL

からロードされます。

MultiSelectionModel

MultiSelectionModel は TreePanel の中での複数選択を可能にします。(TreePanel のデフォルトのセレクションモデルは単一選択のみをサポートします。)

TreeLoader



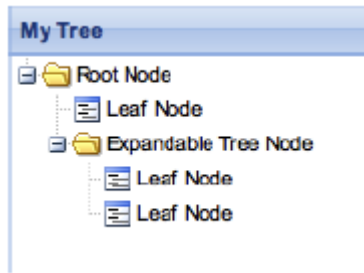
TreeLoader は親ノードが展開されたときに子ノードをロードできるようにします。子ノードはノード定義オブジェクトの配列を返す

URL から受信します。

TreeNode

TreeNode は TreePanel コンポーネントのアイテムです。コンテナノードでは、expandable 属性を enable にします。リーフノードでは、leaf 属性を enable にします。

TreePanel



TreePanel は、階層的なデータを折りたたみ可能なリストに表示するパネルコンポーネントです。少なくとも TreePanel には 1 つの TreeNode がありそれがルートとなります。TreeNode を追加する場合はそのルートノードに追加します。

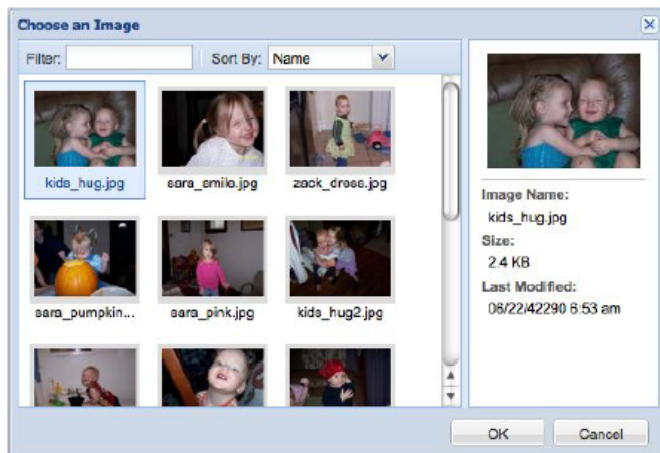
TreePanel は TreeLoader を使って動的にノードデータをロードできます。ノードが展開されたときに、TreeLoader は指定された URL からノードを定義した

配列を自動的に取ってきます。

Views

ストアの動的なデータを表示するには DataView か ListView を使います。ビューを使うと、XTemplate を通してデータのフォーマットとレイアウトを完全にコントロールできます。

DataView



DataView はストアのデータを XTemplate を使ってフォーマットしレイアウトして表示します。割り当てられたストアのデータに変更があったら、DataView は自動的に更新されます。DataView は標準で click, mouseover などの標準アクションや単一選択、複数選択をサポートします。

DataView は BoxComponent なので親のレイアウトによって管理されます。ですが DataView はパネルコンポーネントではないことに注意してください。Toolbar を直接 DataView に配置することはできません。そうしたい場合は Panel でラップして、その Panel に Tool をドッキングします。

ListView

File	Last Modified	Size
kids_hug.jpg	06-22 06:53 am	2.4 KB
sara_smile.jpg	06-22 06:53 am	2.4 KB
zack_dress.jpg	06-22 06:53 am	2.6 KB
sara_pumpkin.jpg	06-22 06:53 am	2.5 KB
sara_pink.jpg	06-22 06:53 am	2.1 KB
kids_hug2.jpg	06-22 06:53 am	2.4 KB
zack_sink.jpg	06-22 06:53 am	2.2 KB
zacks_grill.jpg	06-22 06:53 am	2.8 KB
gangster_zack.jpg	06-22 06:53 am	2.1 KB
zack_hat.jpg	06-22 06:53 am	2.3 KB

ListView はテーブル形式のフォーマットにデータを表示する DataView で、GridPanel によく似ています。

ListView では XTemplate を通してリストのフォーマットやレイアウト全てをコントロールでき、コラムのリサイズやデータ選択を標準でサポートします。しかしながら水平スクロールは実装されていませんし、ドラッグ&ドロップ

やインライン編集のサポートもありません。

🔲 ListViewBooleanColumn

BooleanColumn は論理データを ListView に表示する Column です。

🔲 ListView Column

Column は ListView のコラムでデータがどうレンダリングされるかをコントロールします。生のデータをそのまま表示するのに使います。ExtJS は論理データ, 数値データ, 日付を表示するために特化したタイプの Column も提供します。

🔲 ListView DateColumn

DateColumn は, ListView に日付を表示する Column です。format 属性を設定すると日付の表示形式を指定できます。DateColumn は, Date.js で定義されたフォーマット構文で指定された日付のフォーマットをサポートします。

🔲 ListView NumberColumn

NumberColumn は数値データを ListView に表示する Column です。format 属性を設定すると数値のフォーマットを指定できます。フォーマット文字列は Ext.util.Format.number に従って定義されます。

フォームの構築

Designer は変更の効果をすぐに確認できるので、複雑なフォームのレイアウトを簡単にします。このセクションでは次のことをやってみます。

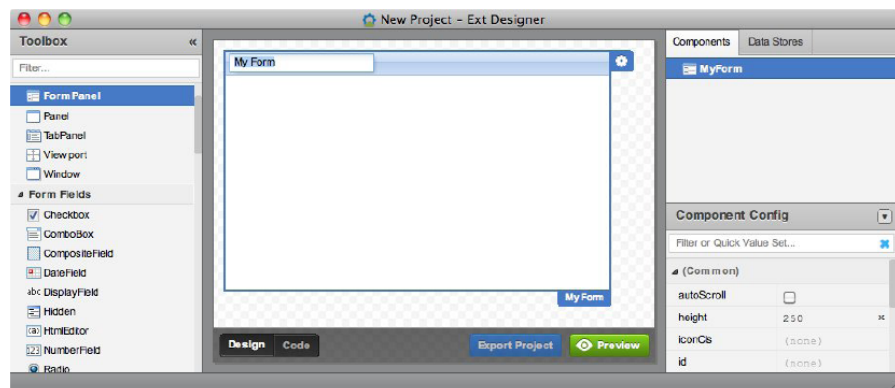
- ◆ Designer で簡単なフォームを構築しフォーム送信のイベントハンドラを割り当てます。
- ◆ ラジオボタンやチェックボックスのグループを追加します。
- ◆ 1行に複数のフィールドを配置して、段組のフォームを作ります。
- ◆ ローカルストアを使った ComboBox を配置します。

簡単なフォームを構築する

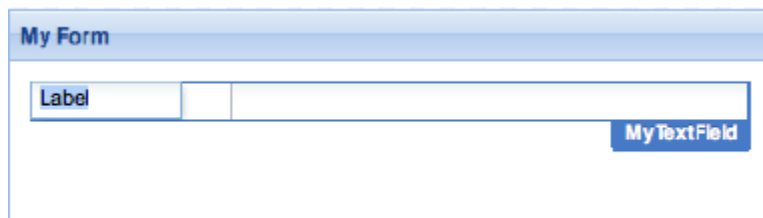
フォームを作るとき、FormPanel コンテナから始めます。FormPanel は追加したすべてのフォームフィールドにラベルを自動的に表示します。フォームの送信とかキャンセルのボタンは通常はフォームのフッターにドッキングされた Toolbar に追加します。Designer でレイアウトが完了したら、プロジェクトをエクスポートし、生成されたコードを編集して送信ボタンとキャンセルボタンにイベントハンドラを割り当てます。

実際に Designer でやってみましょう。

1. Toolbox の FormPanel をダブルクリックすると新しいトップレベルのコンテナがキャンバスに追加されます。デフォルトのタイトルをダブルクリックしてフォームのタイトルを編集できます。パネルの枠をドラッグするとリサイズできます。

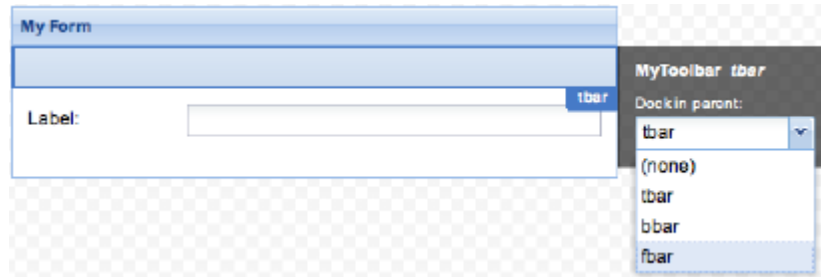


2. 追加したいフィールドを FormPanel に追加します。例えば、TextField をダブルクリックすればテキスト入力フィールドを追加できます。ラベルをダブルクリックすればフィールドラベルを編集できます。

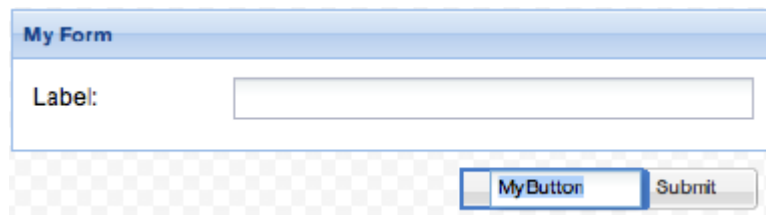


3. 送信ボタンとキャンセルボタン用の Toolbar を FormPanel の中にドラッグします。

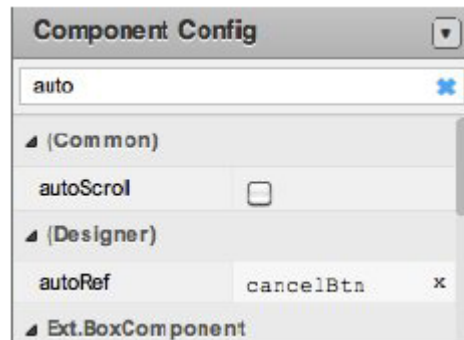
4. その Toolbar のフライアウトコンフィグボタンをクリックし Dock in parent の欄に fbar を指定して、Toolbar を FormPanel のフッタにドッキングさせます。



5. Toolbar にボタンをドラッグして送信とキャンセルのボタンを追加します。ダブルクリックするとボタンのラベルを編集できます。



6. それぞれのボタンの autoRef 属性をセットします (例 : autoRef: submitBtn, autoRef: cancelBtn)。これによりプロジェクトをエクスポートしてイベントハンドラを割り当てるときに簡単にボタンを参照できるようになります。



7. プロジェクトを保存してエクスポートします。
8. 生成されたフォームの.js ファイル (例 : MyForm.js) を編集します。initComponent を呼んでいるところの後に、送信ボタンやキャンセルボタンがクリックされた時に呼び出したい関数を設定します。

```
MyForm = Ext.extend(MyFormUi, {
    initComponent: function() {
        MyForm.superclass.initComponent.call(this);
        this.submitBtn.on('click', this.onSubmitBtnClick, this);
        this.cancelBtn.on('click', this.onCancelBtnClick, this);
    },
});
```

9. フォームクラスに送信とキャンセルのハンドラ関数の実装を追加します。次のように

```
MyForm = Ext.extend(MyFormUi, {
    initComponents: function() {
        MyForm.superclass.initComponents.call(this);
        this.submitBtn.on('click', this.onSubmitBtnClick, this);
        this.cancelBtn.on('click', this.onCancelBtnClick, this);
    },
    onSubmitBtnClick: function() {
        // your implementation here!
    }
    onCancelBtnClick: function() {
        // your implementation here!
    }
});
```

フォームコンポーネントの幅を変更する

FormPanel に配置されたフィールドは自動的に 100%幅に設定されます。フィールドの幅を変更するには次のようにします。

1. anchor 属性をクリアします。
2. width 属性を設定したい幅にピクセル単位で設定します。

パスワードフィールドを追加する

パスワードフィールドを作成するには次のようにします。

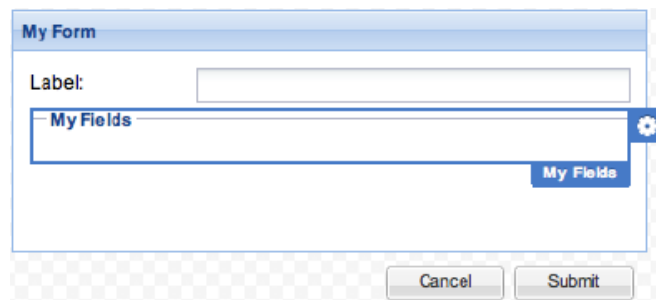
1. テキストフィールドをフォームに追加します。
2. Component Config インспекタでフィールドの inputType 属性に password を設定します。

ラジオボタンやチェックボックスのグループを追加する

Designer でラジオボタンやチェックボックスのグループを作るには、FieldSet コンポーネントにラジオボタンやチェックボックスを追加します。ラジオボタングループの中でひとつだけしか選択できないようにするには、それぞれのボタンの name 属性を同じものにします。

Yes と No の選択をするラジオボタンをフォームに追加するには次のようにします。

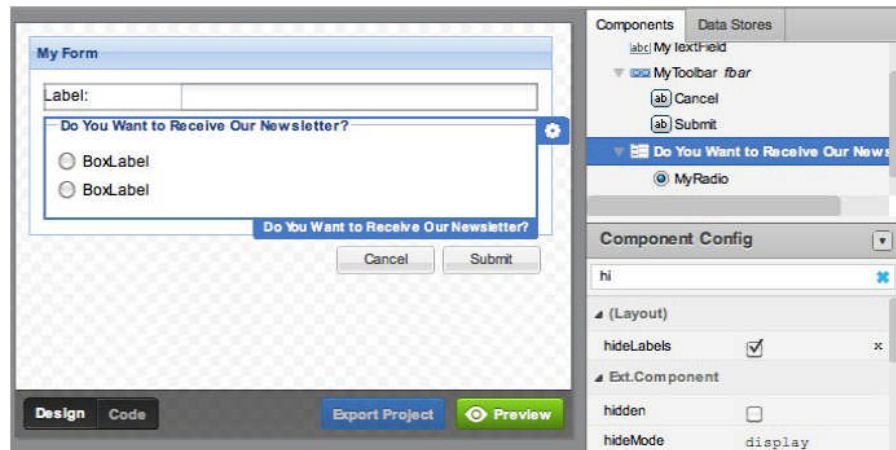
1. FieldSet コンテナを FormPanel にドラッグします。



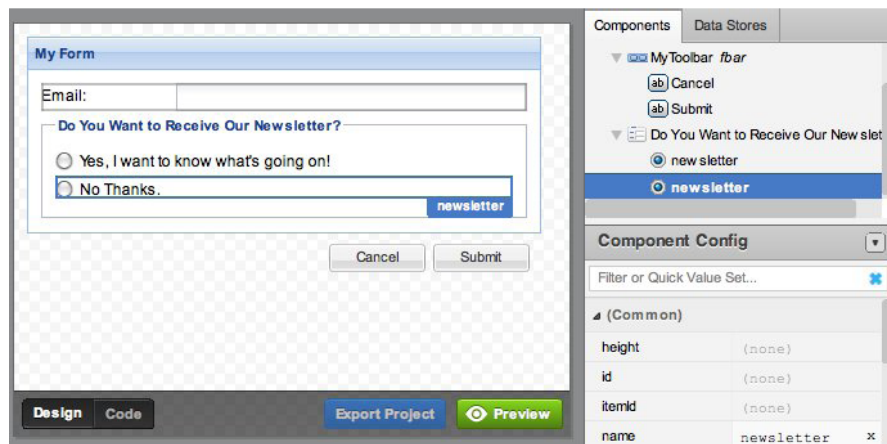
2. 二つの Radio フィールドを FieldSet の中にドラッグします。



3. これでラジオボタンは FieldSet の中に入りました, ラジオボタンの fieldLabel は普通いりませんね。フィールドラベルを隠すには, FieldSet を選択して hideLabels 属性を enable にします。



4. それぞれのラジオボタンの boxLabel をダブルクリックして Yes と No に設定します。
5. 両方のラジオボタンの name 属性を *newsletter* などの同じものに設定します。これにより両方のボタンが同時に選択できなくなります。どちらかのボタンをデフォルトにするには, そのボタンの checked 属性を enable にします。

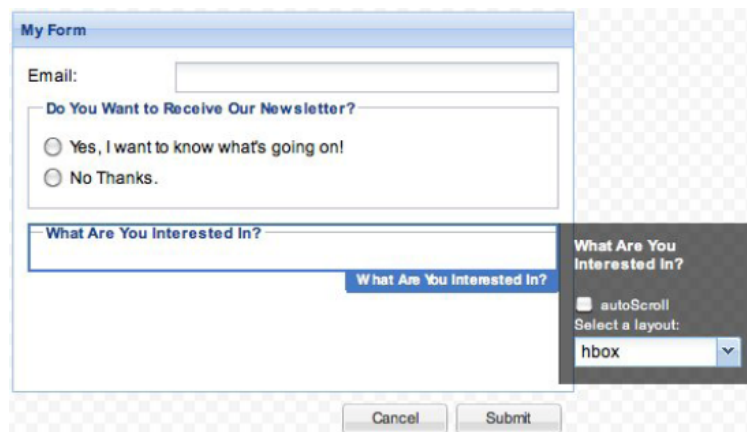


段組でフィールドを配置する

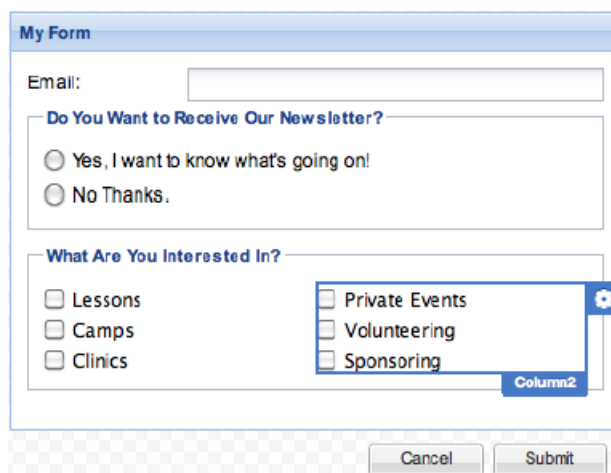
沢山のフィールド数のフォームの場合、縦スクロールを少なくするためにひとつ以上の段組で配置したいことがあります。そうするためには FormPanel の中にネストしてコンテンツを配置します。フォーム全体を複数の段組でレイアウトすることも、一部のセクションだけを段組にすることもできます。

例えば、いくつかの関連するチェックボックスがある場合、スペースを節約するために段組で配置したくなります。

1. チェックボックス用の FieldSet を FormPanel に追加します。
2. FieldSet のレイアウトを hbox にします。



3. それぞれの段用に Container コンポーネントを FieldSet に追加します。そして両方に次の設定をします。段のコンテナを選択するにはキャンバスで選択するよりも、Component リストから選択した方が簡単です。
 - a. レイアウトを vbox にします。
 - b. flex を 1 にします。
 - c. 追加したいチェックボックスに適した高さを height にセットします。例：3 つのチェックボックスを追加するなら 60 ピクセルぐらい。
4. それぞれの段組コンテナにチェックボックスを追加して、boxLabel 属性を設定します。

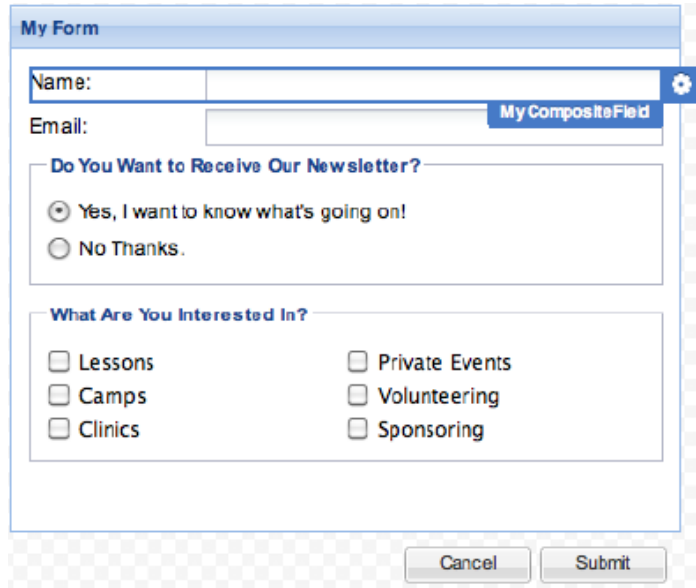


フィールドを水平に並べる

段組を使わずに関連するフィールドの集まりを 1 行に配置したい場合は、CompositeField を使います。

例えば、複数パートの氏名フィールドを作成するには：

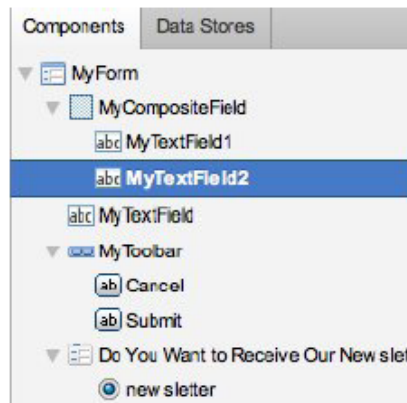
1. CompositeField を FormPanel に追加します。デフォルトでは CompositeField にはひとつの TextField があります。



The screenshot shows a form titled "My Form" with the following elements:

- A "Name:" label followed by a text input field. A blue tooltip labeled "My CompositeField" with a gear icon is positioned over the input field.
- An "Email:" label followed by another text input field.
- A section titled "Do You Want to Receive Our Newsletter?" containing two radio buttons: "Yes, I want to know what's going on!" (selected) and "No Thanks."
- A section titled "What Are You Interested In?" containing six checkboxes arranged in two columns: "Lessons", "Private Events", "Camps", "Volunteering", "Clinics", and "Sponsoring".
- "Cancel" and "Submit" buttons at the bottom.

2. 二つ目の TextField を CompositeField に追加します。キャンバス上にドラッグして CompositeField の上にドロップするか、Component リストにある CompositeField にドラッグします。

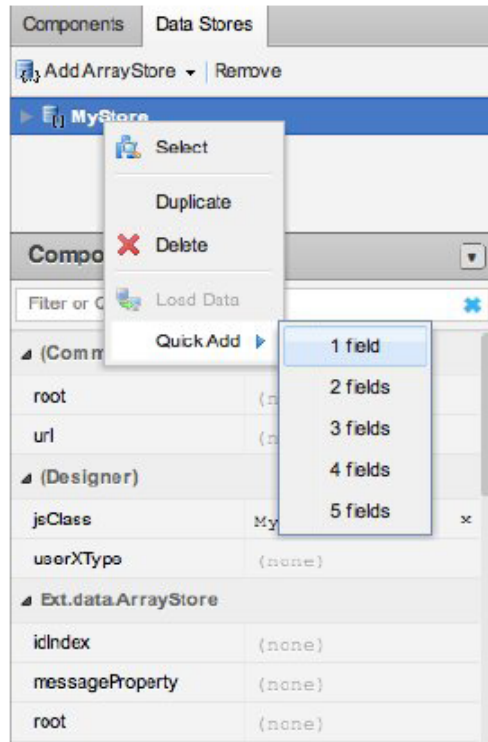


3. それぞれのフィールドの fieldLabel 属性をセットして CompositeField 自身の fieldLabel を削除した場合は、CompositeField のラベルはそれに含まれるフィールドのラベルのリストになります。

ComboBox を配置する

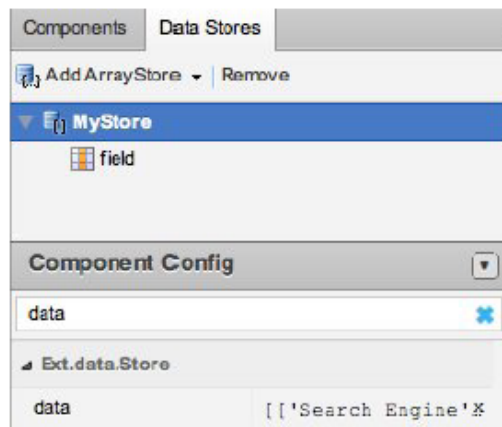
ComboBox にリストされるアイテムはデータストアに定義されます。ComboBox のストアをセットアップする最も簡単な方法は、ローカルの ArrayStore です。

1. Data Store タブから **Add ArrayStore** を選択します。
2. 新しいストアを右クリックして **Add Fields > 1field** を選択します。



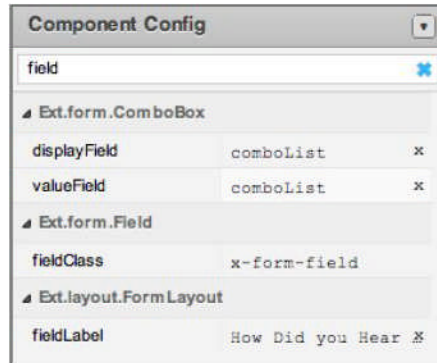
3. フィールド名を設定します。例 : comboList
4. ComboBox に表示したいアイテムのリストを設定します。ストアの data 属性の中に配列の配列として設定します。

例 : [['Search Engine'], ['Online Ad'], ['Facebook']]

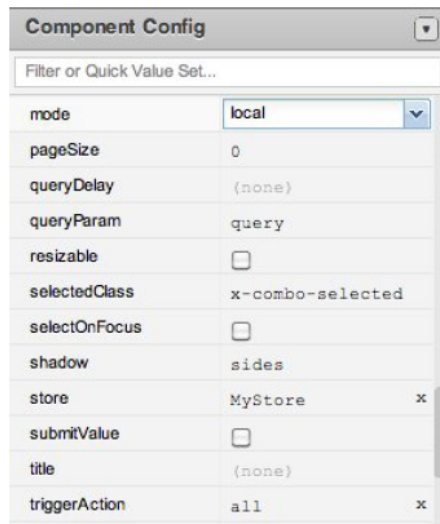


配置した ComboBox にローカル ArrayStore を使うには次の様にします。

1. ComboBox のフライアウトコンフィグボタンでストアを選択します。
2. ComboBox を設定して、ストアのどのフィールド displayField と valueField として使用するかを指定します。



3. スタアのアイテム全てをドロップダウンリストに表示するために、triggerAction 属性を all にします。mode 属性を local に設定します。



Tip : フィールドラベルの幅をコントロールするには、labelStyle 属性に幅をセットすることができます。

メニューを構築する

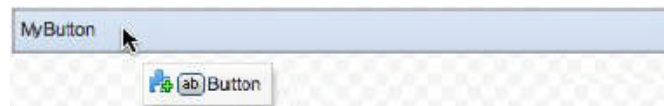
Button に Menu コンポーネントを追加して、そこに Menu アイテムを加えればメニューに変えることができます。Designer でメニューを設定したら、プロジェクトをエクスポートして生成されたコードを編集してそれぞれのメニューアイテムのイベントハンドラを割り当てます。

メニューバーの構築

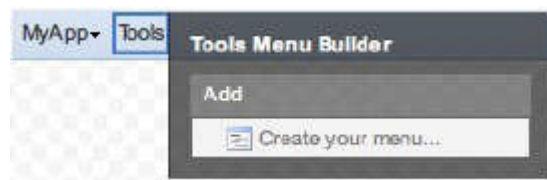
よくあるメニューバーを作るには Toolbar から始め、それぞれのメニューの Button を追加し、それぞれのボタンの Menu コンポーネントを設定します。

次は Designer での作業の例です。

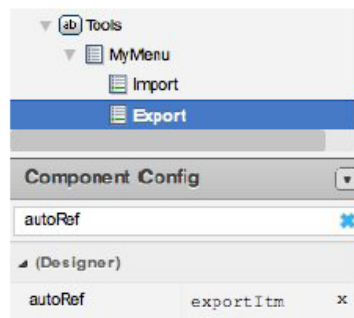
1. Toolbar をキャンバスにドラッグします。
2. それぞれのメニュー用の Button を Toolbar の上にドラッグします。ここでは 2 つのボタンを追加します。



3. ボタンのラベルをそれぞれのメニューの名前に設定します。例：MyApp と Tools
4. それぞれのボタンに Menu コンポーネントを追加し 2 つのメニューアイテムを追加します。
 - a. ボタンのフライアウトコンフィグボタンをクリックして Add を選択すると Menu コンポーネントを追加します。これで Menu Item も自動的に追加されます。



- b. Add を再度クリックして二つ目のアイテムをメニューに追加します。
ボタンの上に Menu をドラッグしてもメニューコンポーネントを追加できます。同様に Menu Item, CheckItem, Separator, TextItem コンポーネントをメニューにドラッグして追加できます。
5. 各メニューアイテムの text 属性にメニューに表示したい名前をセットします。例：MyApp のアイテムに About と Preferences を、Tools のアイテムに Import と Export をセットします。
6. 各メニューアイテムの autoRef 属性を設定します。例：Export メニューアイテムの autoRef を exportItm にします。これによりプロジェクトをエクスポートしたときに、イベントハンドラを割り当てるためにアイテムの参照が簡単になります。



7. メニューバーの設定が終わったら、プロジェクトを保存してエクスポートします。
8. 生成されたメニューバーの .js ファイル（例：MyToolbar.js）を編集します。
initComponent を呼び出しているところの後に、メニューアイテムがクリックされたときに呼び出したい関数を設定します。

```

MyToolbar1Ui = Ext.extend(Ext.Toolbar, {
    width: 400,
    initComponent: function() {
        this.items = [
            {
                xtype: 'button',
                text: 'MyApp',
                menu: {
                    xtype: 'menu',
                    items: [
                        {
                            xtype: 'menuitem',
                            text: 'About'
                        },
                        {
                            xtype: 'menuitem',
                            text: 'Preferences'
                        }
                    ]
                }
            },
            {
                xtype: 'button',
                text: 'Tools',
                menu: {
                    xtype: 'menu',
                    items: [
                        {
                            xtype: 'menuitem',
                            text: 'Import'
                        },
                        {
                            xtype: 'menuitem',
                            text: 'Export',
                            ref: '../../exportItm'
                        }
                    ]
                }
            }
        ];
        MyToolbar1Ui.superclass.initComponent.call(this);
        this.aboutItm.on('click', this.onAboutItemClick, this);
        this.prefItm.on('click', this.onPrefItmClick, this);
        this.importItm.on('click', this.onImportItmClick, this);
        this.exportItm.on('click', this.onExportItmClick, this);
    }
});
    
```

9. ツールバークラスのハンドラ関数の実装を追加します。例：

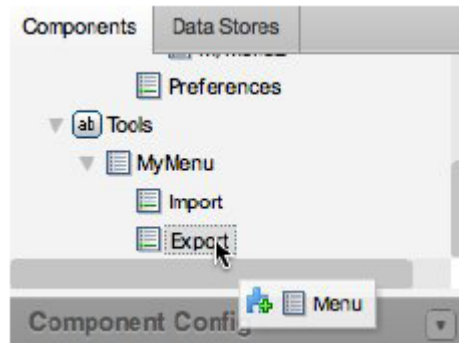
```
MyToolBarUi = Ext.extend(Ext.Toolbar, {
    width: 400,
    initComponents: function() {
        .
        .
        .
        MyToolBarUi.superclass.initComponent.call(this);
        this.aboutItm.on('click', this.onAboutItmClick, this);
        this.prefItm.on('click', this.onPrefItmClick, this);
        this.importItm.on('click', this.onImportItmClick, this);
        this.exportItm.on('click', this.onExportItmClick, this);
    }
    onAboutItmClick: function() {
        // your implementation here!
    }
    onPrefItmClick: function() {
        // your implementation here!
    }
    onImportItmClick: function() {
        // your implementation here!
    }
    onExportItmClick: function() {
        // your implementation here!
    }
});
```

サブメニューを作る

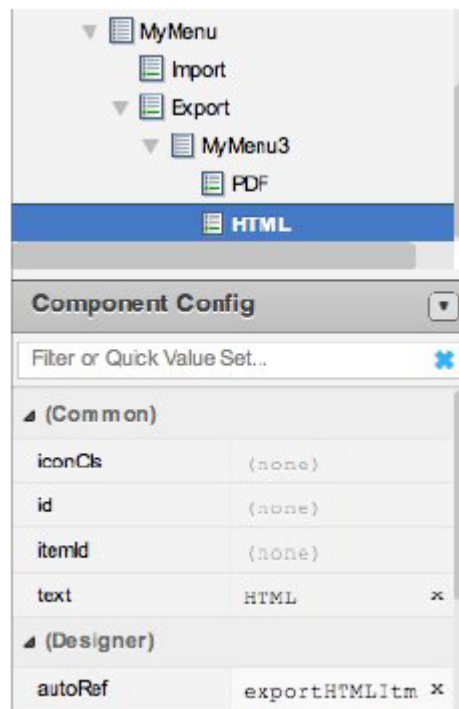
Designer でサブメニューを作るのは簡単です。Menu コンポーネントを既存のメニューアイテムに追加してサブメニューのアイテムを追加します。イベントハンドラをサブメニューアイテムに割り当てるのは、他のメニューアイテムにハンドラを割り当てるのと全く同じです。

例として、Export メニューアイテムにサブメニューを追加します。

1. Menu コンポーネントを Export メニューアイテムの上にドラッグします。キャンバスの上でも Component タブでもどちらでもできます。



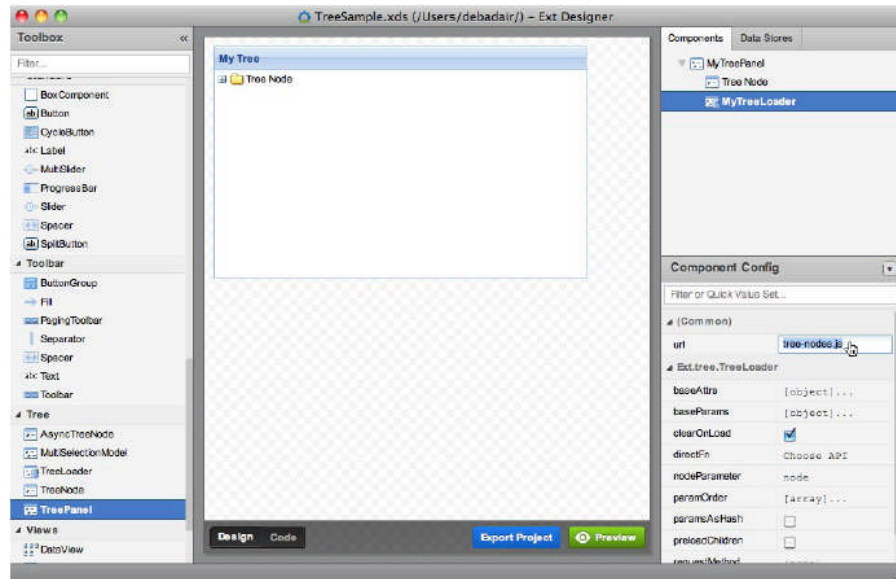
2. 使用したい各サブメニューアイテム用に Menu Item に追加します。この例では PDF と HTML を追加します。
3. text 属性と autoRef 属性をそれぞれ設定します。例：text : HTML と autoRef : exportHTMLitm



ツリーを配置する

キャンバスに TreePanel を追加すると Designer は自動的にルートノードと TreeLoader を追加します。ツリーを配置するために TreeLoader を使うには次のようにします。

1. TreeLoader の url 属性をノード定義を受信したい場所を指すように設定します。
注：この場所は Project Settings で設定された URL Prefix からの相対アドレスを指定します。URL Prefix を参照, 変更するには Designer の Edit メニューから Edit Preference を選択します。



2. ノードを定義します。ノードが展開されたとき、サーバーリクエストが送られ子ノードが TreeLoader で指定された URL からロードされます。そのレスポンスは各エレメントがノード定義オブジェクトである次のような Javascript の配列定義でなければなりません。

```

tree-nodes.js
Last Saved: 8/5/10 3:05:38 PM
File Path: ~/Sites/Examples/tree-nodes.js

[
  {
    id: 1,
    text: 'A leaf Node',
    leaf: true
  },
  {
    id: 2,
    text: 'A folder Node',
    children: [
      {
        id: 3,
        text: 'A child Node',
        leaf: true
      }
    ]
  }
]

```