Customizing Touch UI Dialog Fields

CQ Gems on AEM
October 7th, 2015

Basics

Touch-optimized UI

Next generation User Interface

Optimized for **Touch**, with Desktop in mind

Meant to be highly **customizable**

Based on **Coral UI** + **Granite UI**

Adobe

Touch-optimized UI

Next generation User Interface

Optimized for **Touch**, with Desktop in mind

Meant to be highly **customizable**

Based on **Coral UI** + **Granite UI**

Adobe

Coral UI

Widget library (CSS / JS)

Consistent UX across all cloud solutions

Adobe

Granite UI

Foundational building blocks

Coral UI markup wrapped into Sling components

Components for building UI consoles and **dialogs**

Adobe

Granite UI

Foundational building blocks

Coral UI markup wrapped into Sling components

Components for building UI consoles and **dialogs**

Adobe

Component Dialogs

To edit properties of an component instance

Adobe

Classic UI vs. Touch UI

Dialogs

Adobe

Classic UI Dialogs

`dialog` node

ExtJS widgets / xtypes

rendered client-side

Adobe

Classic UI Dialogs

`dialog` node

ExtJS widgets / xtypes

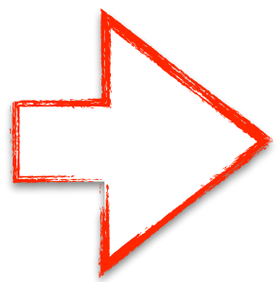
rendered client-side

Adobe

Classic UI Dialogs

`dialog` node

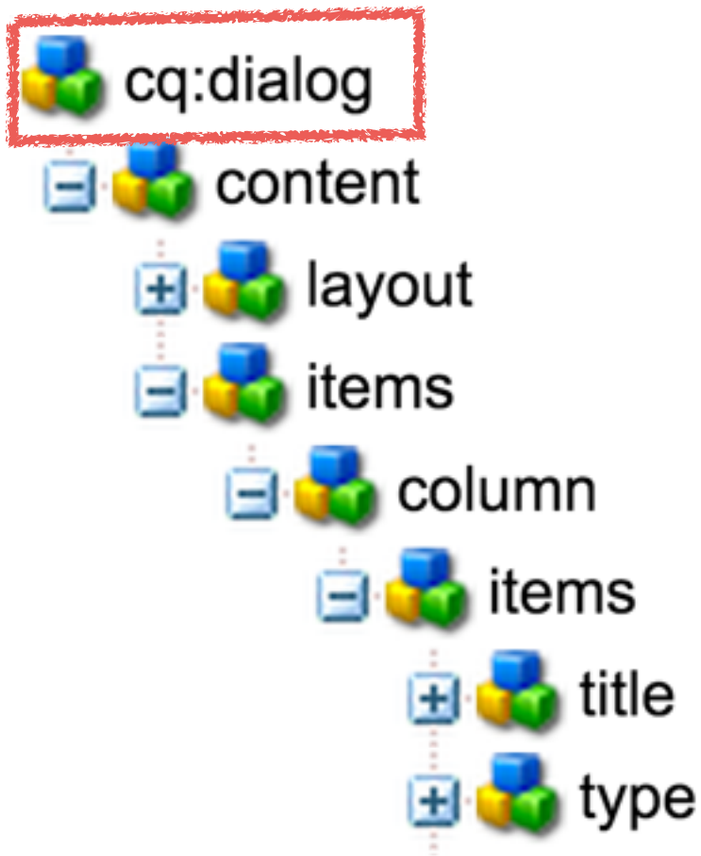
ExtJS widgets / xtypes



rendered client-side

Adobe

Touch UI Dialogs



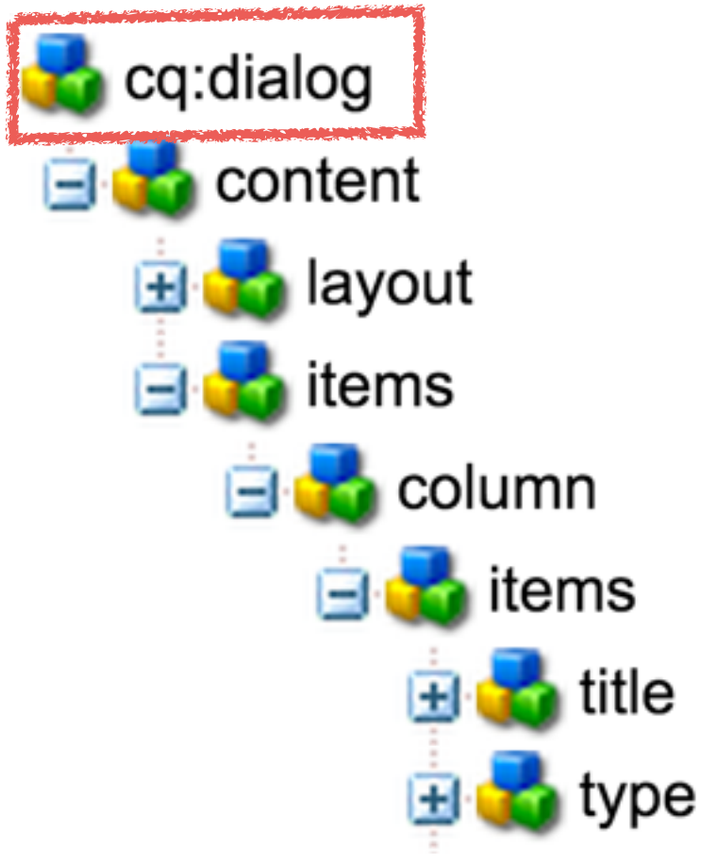
`cq:dialog` node

Granite UI resource types

rendered server-side

Adobe

Touch UI Dialogs



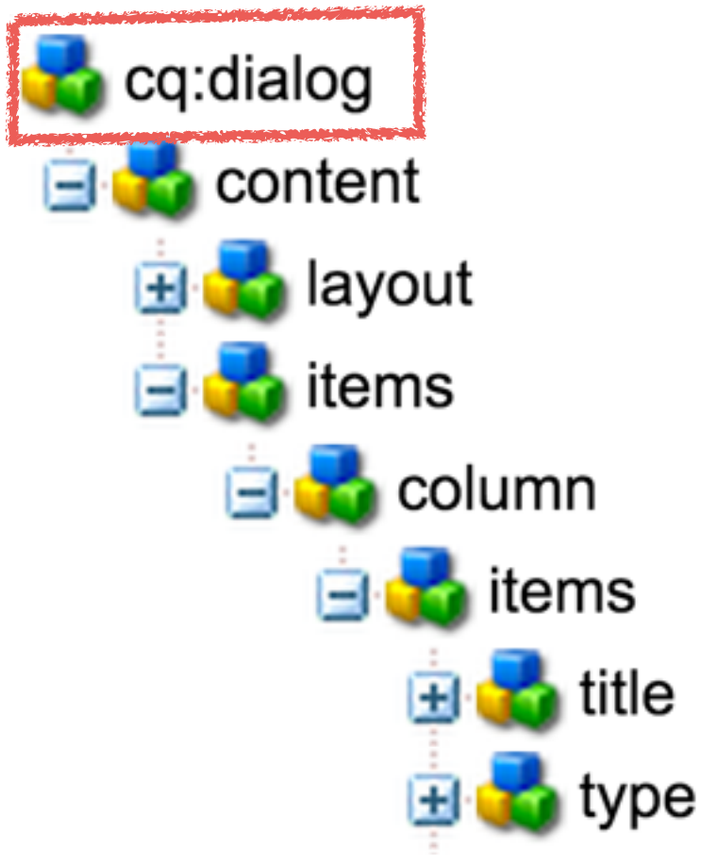
`cq:dialog` node

Granite UI resource types

rendered server-side

Adobe

Touch UI Dialogs



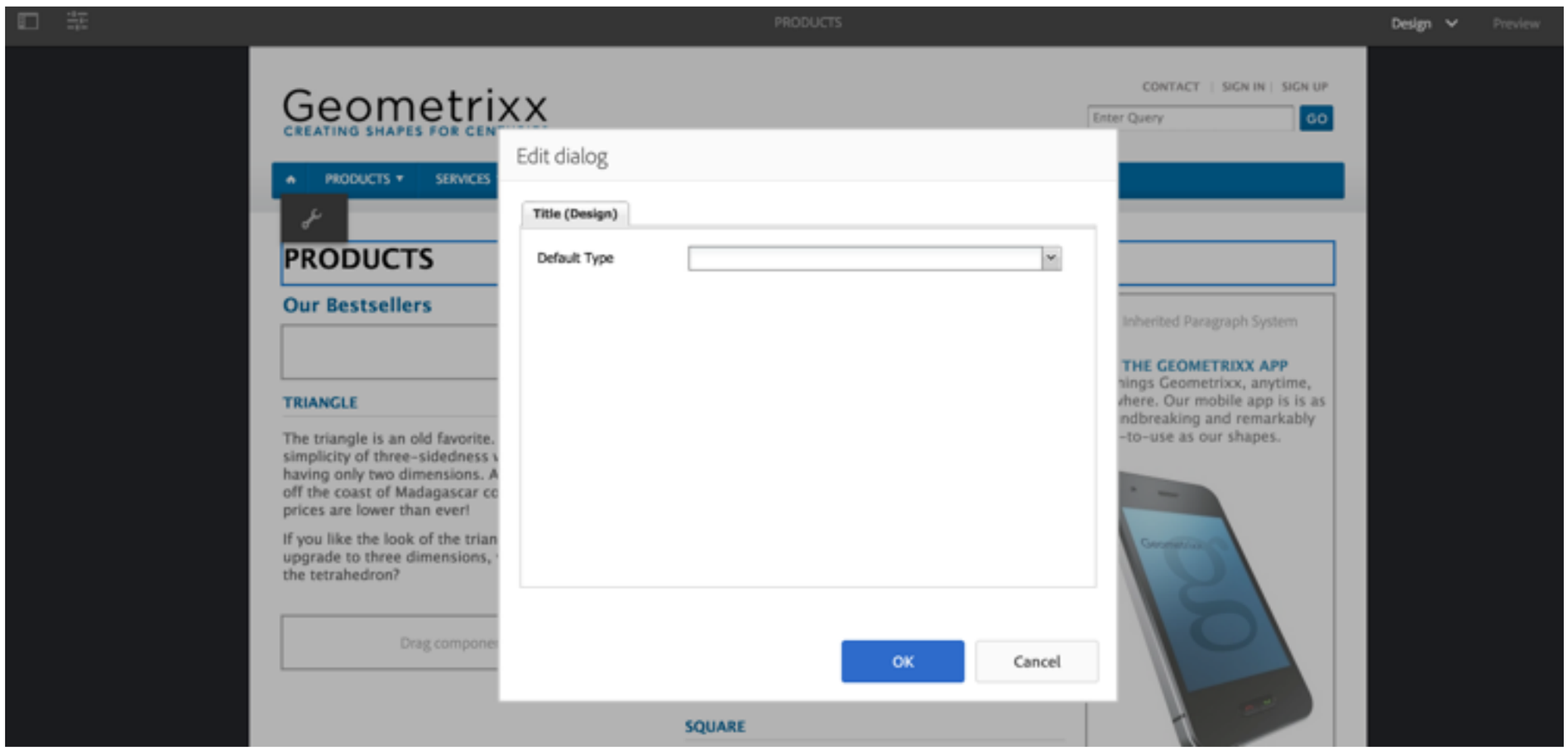
`cq:dialog` node

Granite UI resource types

rendered server-side

Adobe

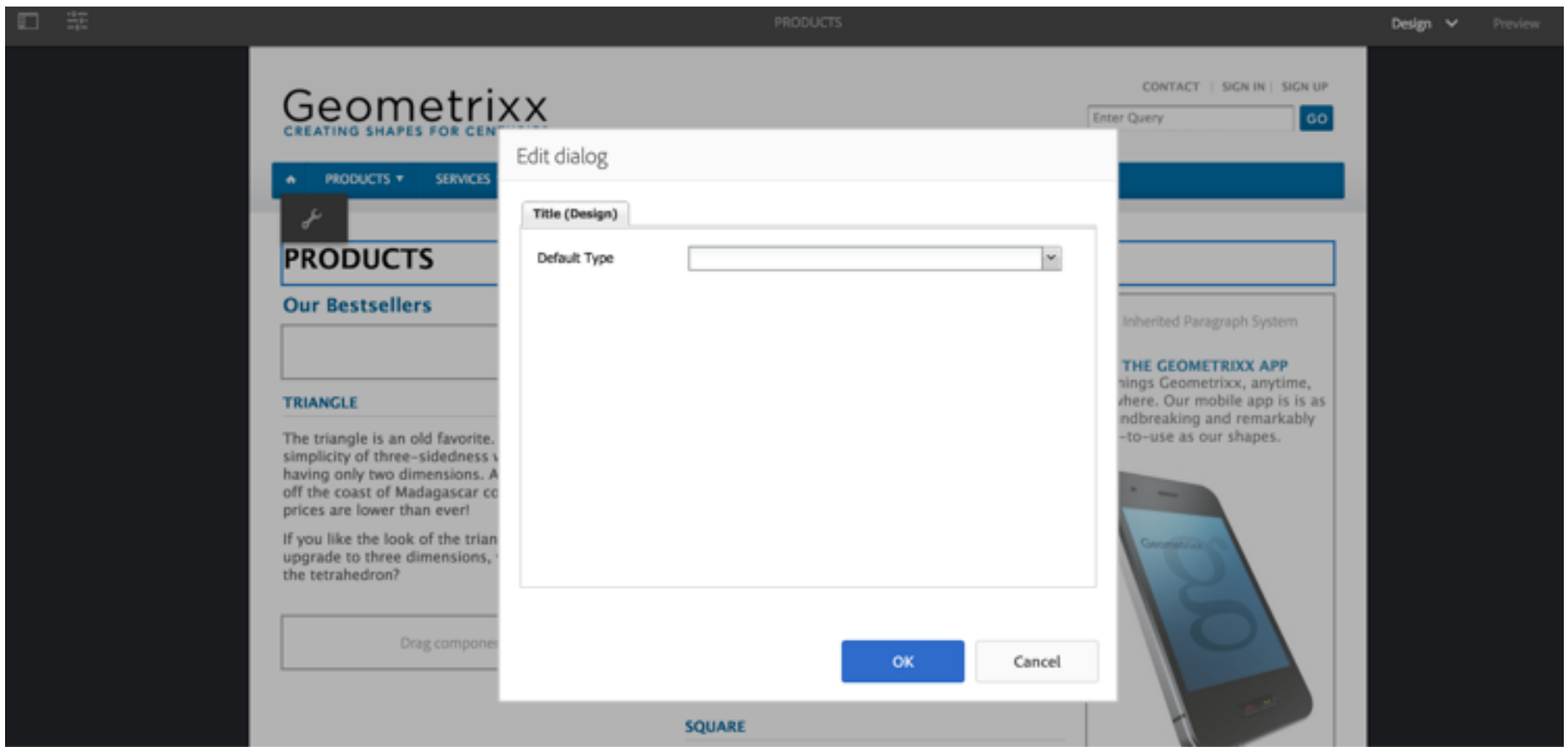
Classic UI Dialogs in Touch UI?



Compatibility layer when **no Touch UI** dialog defined

Adobe

Classic UI Dialogs in Touch UI?



Migration: Dialog converter tool

Adobe

Examples

Classic UI: /libs/foundation/components/title/dialog

Touch UI: /libs/foundation/components/title/cq:dialog

Adobe

Creating custom Dialog Fields

Adobe

Creating a new field

0) Check `granite/ui/components/foundation/form/`

1) Create Resource type to render markup

2) Create Client library to define style / behavior for markup

Adobe

Creating a new field

0) Check `granite/ui/components/foundation/form/`

1) Create Resource type to render markup

2) Create Client library to define style / behavior for markup

Adobe

Creating a new field

1) Resource type to render **markup**

= usual **component development**

Adobe

Creating a new field

1) Resource type to render **markup**

inherit from Granite UI base **field**

```
sling:resourceSuperType=
    “granite/ui/components/foundation/form/field”
```


Adobe

Creating a new field

1) Resource type to render **markup**

inherit from Granite UI base **field**

override **render.jsp**

Adobe

Creating a new field

1) Resource type to render **markup**

inherit from Granite UI base **field**

override **render.jsp**

contract: read and display **form value** from request

```
// Gives you the value of the field (read from the content)
ValueMap vm = (ValueMap) request.getAttribute(Field.class.getName());
vm.get(“value", String.class);
```


Adobe

Examples

cqgems/customizingfield/components/colorpicker

granite/ui/components/foundation/form

Adobe

Creating a new field

2) Client library to define **style** / **behavior**

= usual **client library** development

Adobe

Creating a new field

2) Client library to define **style** / **behavior**

generic field ? use **`cq.authoring.dialog`** as category

specific field ? use **`extraClientLibs`** property on dialog

Adobe

Creating a new field

2) Client library to define **style** / **behavior**

generic field ? use **`cq.authoring.dialog`** as category

specific field ? use **`extraClientLibs`** property on dialog

Adobe

Example

cqgems/customizingfield/components/colorpicker/clientlibs

Adobe

Event handling on Dialog Fields

Adobe

Handling events on fields

ExtJS listeners replacement

Listeners in custom Client library instead

Mark your custom field with **CSS class**

Hook JS event listener using that CSS class

Adobe

Handling events on fields

ExtJS listeners replacement (**avoid listeners in content!**)

Listeners in custom Client library instead

Mark your custom field with **CSS class**

Hook JS event listener using that CSS class

Adobe

Handling events on fields

ExtJS listeners replacement (**avoid listeners in content!**)

Listeners in custom Client library instead

Mark your custom field with **CSS class**

Hook JS event listener using that CSS class

Adobe

Handling events on fields

ExtJS listeners replacement (**avoid listeners in content!**)

Listeners in custom Client library instead

Mark your custom field with **CSS class**

Hook JS event listener using that CSS class

Adobe

Handling events on fields

ExtJS listeners replacement (**avoid listeners in content!**)

Listeners in custom Client library instead

Mark your custom field with **CSS class**

Hook JS event listener using that CSS class

Adobe

Handling events on fields

ExtJS listeners replacement (**avoid listeners in content!**)

Listeners in custom Client library instead

Mark your custom field with **CSS class**

Hook JS event listener using that CSS class

(see CoralUI documentation for API / Events)

Adobe

Example

cqgems/customizingfield/components/clientlibs/customizingfield

Adobe

Validation for Dialog Fields

Adobe

Handling validation on fields

Mandatory?

set **`required`** property on node

Advanced validation?

set **`validation`** property (as a key to a registered validator)

see Granite UI Validation API / jQuery Validator

Adobe

Handling validation on fields

Mandatory?

set **`required`** property on node

Advanced validation?

set **`validation`** property (as a key to a registered validator)

use Granite UI Validation API

Adobe

Examples

cqgems/customizingfield/components/clientlibs/customizingfield/js/validations.js

cq/gui/components/authoring/dialog/clientlibs/dialog/js/validations.js

Adobe

Resource type is an abstraction

Adobe

Resource type is an abstraction

Resource type = **semantic intention**

Look and feel = implementation detail

Even though the look and feel usually changes over time,

the intention stays the time

Adobe

Resource type is an abstraction

Resource type = **semantic intention**

Look and feel = implementation detail

Content structure **declares** intentions,
resource type **implement** them.

Adobe

Example

cqgems/customizingfield/components/colorpicker2

Adobe

Summary

Classic UI / Touch UI Dialogs

Creating Fields to be used in Dialogs

Fields event handling, validation

Adobe

Resources

Touch UI Concepts

Granite UI Reference

- Granite UI Form Fields Reference

Coral UI Reference

Adobe