COMPONENTS IN AEM :-   
structure:-

cq:dialog

-----content

---- items

----tabs

------items

------properties

-----items

----columns

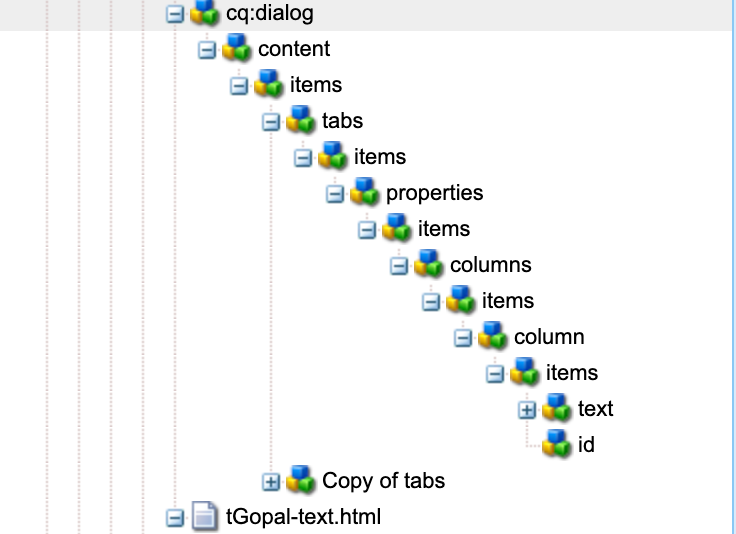
----items

----column

----items

---text

---id



Different properties present in AEM :-

Text - libs/cq/gui/components/authoring/dialog/richtext

Title - libs/

File reference

Link

Linkpath

How many UI’s(user interfaces are there in aem) ?:-

· **Classic UI**

· **Touch UI**

· **Coral UI**

· **Granite UI**

### 1. Classic UI

**Classic UI:**

* **Description:** The Classic UI is the older user interface of AEM, designed primarily for desktop use. It uses ExtJS for its components and layouts.
* **Storage:** Classic UI configurations and components are typically stored in the /libs and /apps directories within AEM.
* **Real-time Example:**
  + **Example:** Using the Classic UI, a content author can edit page content using the traditional WYSIWYG editor and sidekick panel.
  + **Usage:** This UI was widely used in AEM versions prior to AEM 6.0 and is now considered deprecated, with most modern implementations transitioning to Touch UI.

### 2. Touch UI

**Touch UI:**

* **Description:** The Touch UI is the modern, responsive, and touch-friendly interface introduced in AEM 5.6. It is designed for both desktop and mobile devices, providing a more intuitive and flexible authoring experience.
* **Storage:** Touch UI components and configurations are also stored in the /libs and /apps directories, specifically under paths like /libs/cq/gui/components/authoring/dialog
* **Real-time Example:**
  + **Example:** A content author uses the Touch UI to create and edit a webpage by dragging and dropping components onto the page, and using inline editing features.
  + **Usage:** This UI is the standard for current AEM implementations, offering improved usability and performance on various devices.

### 3. Coral UI

**Coral UI:**

* **Description:** Coral UI is a set of standardized, reusable UI components based on the Adobe Spectrum design system. It is used within the Touch UI to ensure consistency and modern aesthetics across the AEM interface.
* **Storage:** Coral UI resources are typically located within the /libs/clientlibs directory in AEM.
* **Real-time Example:**
  + **Example:** A developer creates a custom dialog for a new component using Coral UI fields like text inputs, dropdowns, and date pickers to ensure a consistent look and feel.
  + **Usage:** Coral UI components are used throughout the Touch UI to build dialogs, forms, and other interactive elements.

### 4. Granite UI

**Granite UI:**

* **Description:** Granite UI is the framework that underpins the Touch UI, providing the backend logic and client-side libraries necessary to build complex user interfaces in AEM. It leverages Coral UI components for visual elements.
* **Storage:** Granite UI resources and configurations are stored under /libs/granite and /apps/<your-project>/granite.
* **Real-time Example:**
  + **Example:** Implementing a multi-step form for content authors to upload assets and enter metadata, utilizing Granite UI components for form fields, validation, and submission logic.
  + **Usage:** Granite UI is used to create the structural and interactive foundation of the Touch UI, integrating backend functionality with frontend presentation.

### Where UIs are Stored

**Classic UI:**

* + Stored primarily in /libs and /apps under paths related to the Classic UI, such as /libs/cq/ui/classic.

**Touch UI:**

* + Components and configurations are found under /libs/cq/gui/components and /apps/<your-project>/components.

**Coral UI:**

* + Resources like CSS, JS, and HTML templates are located in /libs/clientlibs and specific Coral UI paths like /libs/coral/ui.

**Granite UI:**

* + Found under /libs/granite and /apps/<your-project>/granite, including server-side and client-side resources.

### Real-time Example for Each UI:

**Classic UI:**

* + **Example:** A marketer logs into the Classic UI to edit a product page using the traditional sidekick and dialog-based editing interface. This involves modifying text, images, and page properties through the Classic UI’s structured editor.

**Touch UI:**

* + **Example:** A content author uses the Touch UI to build a new landing page by dragging components from the side rail onto the page, editing content inline, and configuring component properties through modern, responsive dialogs.

**Coral UI:**

* + **Example:** A developer customizes a Touch UI dialog for a new banner component, using Coral UI elements like coral-textfield, coral-datepicker, and coral-select to ensure consistent design and behavior.

**Granite UI:**

* + **Example:** An administrator sets up a complex workflow using Granite UI to configure the workflow steps, user assignments, and notifications, leveraging Granite UI's extensible framework and components for backend processing and frontend display.

/libs/granite/ui/components/coral/foundation/container

cq/gui/components/authoring/dialog