

## TASK-1 CODE

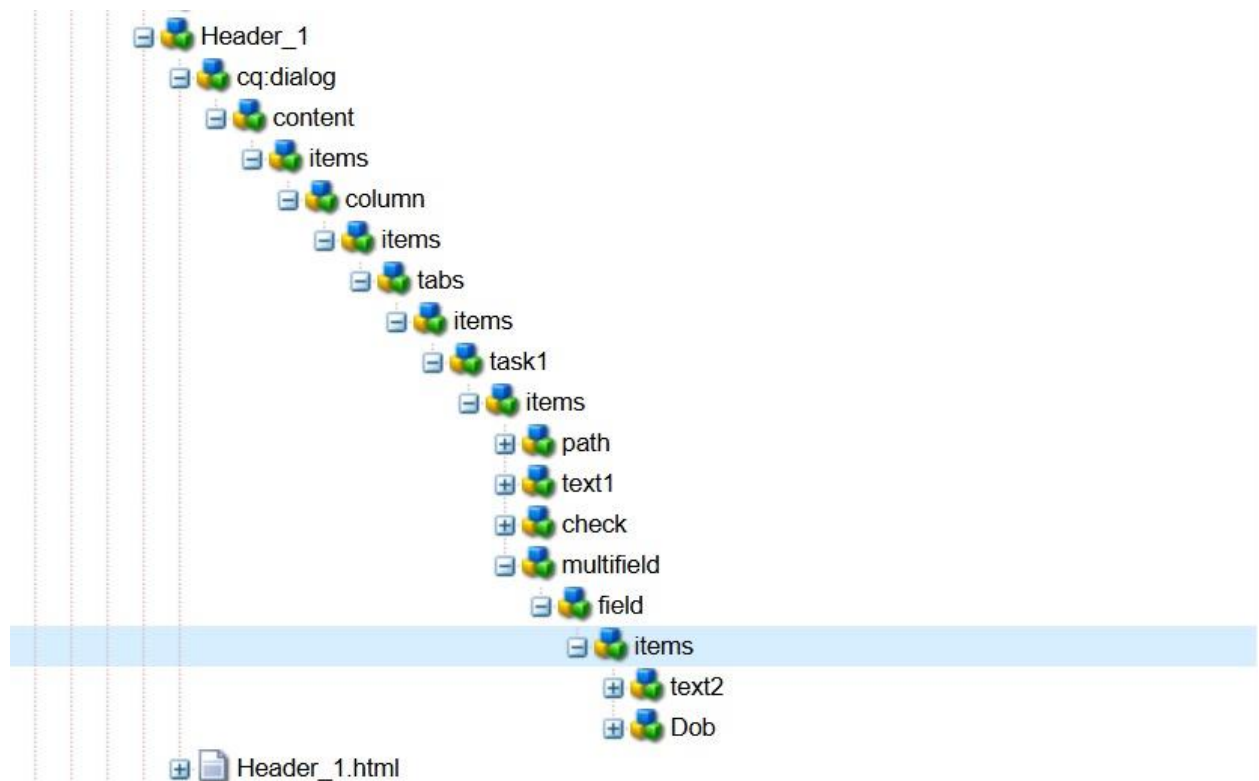
## QUESTION

- 1- Create a component Named Header with Path field, text field, multifield (text field & Date picker) and checkbox.
- 2- Render the authored field via sling model on page using HTL

## 1. Create a Component Named Header

### Steps:

1. **Define the Component:**
  - In your AEM project, create a new component named "Header."
  - Define the component properties and dialog structure.
2. **Dialog Configuration:**
  - Create a dialog for the Header component.
  - Add the required fields (path field, text field, multifield with text field and date picker, checkbox) to the dialog.



## . Render the Authored Field via Sling Model on Page Using HTL

- **Objective:** Render the data authored in the Header component using a Sling Model and HTL (HTML Template Language).
- **Steps:**
  1. **Create a Sling Model:**
    - Develop a Sling Model to map the data from the Header component.
    - Annotate the model with `@Model` and map the properties.

```
package com.adobe.aem.guides.demo.core.models;

import java.util.List;

import javax.inject.Inject;

import org.apache.sling.api.resource.Resource;

import org.apache.sling.models.annotations.DefaultInjectionStrategy;

import org.apache.sling.models.annotations.Model;

import org.apache.sling.models.annotations.injectorspecific.ChildResource;

@Model(adaptable = Resource.class, defaultInjectionStrategy =
DefaultInjectionStrategy.OPTIONAL)

public class Task_2 {

    @Inject

    private String pathfield;

    @Inject

    private String textfield;

    @Inject

    private String checkbox;

    @ChildResource

    private List<Unit>multi;
```

```

    public List<Unit> getMulti() {

        return multi;

    }

    public String getPathfield() {

        return pathfield;

    }

    public String getTextfield() {

        return textfield;

    }

    public String getCheckbox() {

        return checkbox;

    }
}

```

## Multifield model

```

package com.adobe.aem.guides.demo.core.models;

import javax.inject.Inject;

import org.apache.sling.api.resource.Resource;

import org.apache.sling.models.annotations.DefaultInjectionStrategy;

import org.apache.sling.models.annotations.Model;

@Model(adaptables = Resource.class,defaultInjectionStrategy = DefaultInjectionStrategy.OPTIONAL)

public class Unit {

    @Inject

    private String text2;

    @Inject

```

```

private String dob;

public String getText2() {

    return text2;

}

public String getDob() {

    return dob;

}}

```

## Render Data Using HTL:

- Use the Sling Model in your HTL file to render the Header component data.

```

<h2 class="sknv"> Header_1 component</h2>

<div data-sly-use.obj="com.adobe.aem.guides.demo.core.models.Task_2">

    <h3>PathField :${obj.pathfield}</h3>

    <h3 class="my">TextField :${obj.textfield}</h3>

    <h3>checkBox :${obj.checkbox}</h3>

    <div data-sly-list.object="${obj.multi}">

        <h3>Text :${object.text2}</h3>

        <h3>DateOfBirth :${object.dob}</h3>

    </div>

</div>

</div>

<sly data-sly-use.clientlib="/libs/granite/sightly/templates/clientlib.html">

    <sly data-sly-call="${clientlib.all @ categories='libraryclient'}"/>

</sly>

```

