


How Headless Adaptive Form works?

A Headless Adaptive Form is essentially a JSON structure (schema) consisting form fields (Text box, choices, and many more fields) and corresponding rules (conditional logic) to add interactive behavior to the form. You can use REST APIs in your application or website to request the hosted JSON structure and natively render the JSON structure as a form in your app or website. A single Headless Adaptive Form can serve multiple webpages and applications without making any app or website-specific changes to it.

 How Headless Adaptive Form works

Architecture {#architecture}

A typical Headless Adaptive Forms architecture constitutes an Adobe Experience Manager Forms Server, Headless Adaptive Forms hosted on Adobe Experience Manager Forms Server, and your frontend apps (Website, mobile application, JavaScript apps, chat applications, and more) for channel-specific form renditions.

Typical architecture of a Headless Adaptive Forms deployment looks like the following:

 Architecture

Component of a Headless Adaptive Forms implementation

Adobe Experience Manager Server: Along with serving as host for Headless Adaptive Forms, Adobe Experience Manager provides the following backend capabilities:

- RESTful APIs to list, fetch, pre-fill, validate, submit, track submission status of headless forms.
- Visual editor to easily develop a Headless Adaptive Forms.
- Forms Data Model to receive or send data to disparate data sources.
- Workflow engine to automate complex tasks.

Headless Adaptive Forms: A Headless Adaptive Form is represented as a .json file. The JSON structure defines components, constraints, and structure of a form.

Front-end Apps: Front-end apps like, SPA (Single Page Applications), Mobile Apps, JavaScript Apps, consume Headless Adaptive Forms (the JSON Form Representation) and render the form on a client. You can use the React renderer component shipped with Headless Adaptive Forms to render an Adaptive Form or build your own custom component to natively render Headless Adaptive Forms.

Understanding Headless Adaptive Forms definition

<Section Under Development>

Developer tools

In a typical development cycle, you start with creating and hosting Headless Adaptive Forms on Adobe Experience Manager Forms Server. In the second step, you map your UI components or use a public UI components library, such as Google Material UI or Chakra UI to style your forms. In the last step, you fetch

and display Headless Adaptive Forms in your application (Website, mobile application, JavaScript apps, chat applications, or many other surfaces).

The following tools help you create and integrate Headless Adaptive Forms to your applications:

Forms Web SDK (Client-Side Runtime): Forms Web SDK is a client-side JavaScript library. It allows you to apply client-side validations on form fields, maintain state of the form, and provides hooks to connect form with UI layer or adaptive forms rendered component. It allows customers to validate constraints applied to various fields of a form and hooks to connect JSON structure of form to the UI framework. The Forms Web SDK has the following components:

- **Business rule processor:** The business rule processor accepts the forms JSON structure as input, manages the state of the form fields, executes rules, and event handlers present in the JSON.
- **React binder:** Provides hooks over controller to add state to Form Components. It is also helpful in pre-filling a form.
- **Components library:** It provides React Spectrum Components and uses hooks in React Binder module to add state to those components.

Along with providing the APIs to validate constraints applied to various fields of a form, Forms Web SDK provides hooks to connect Headless Adaptive Forms to the UI framework. It also provides React Renderer for Headless Adaptive Forms to help integrate a Headless Adaptive Form to your application. The following components of Web SDK are available:

- [@aemforms/af-react-components](#)
- [@aemforms/af-react-renderer](#)
- [@aemforms/af-core](#)

All of these components are included in AEM Archetype. When you create an AEM Archetype 37 or later project for Headless Adaptive Forms, the latest version of above listed libraries is included in the project.

Started Application: Adobe has also released a started application to help you start quickly with Headless Adaptive Forms.

Storybook: [Storybook](#) provides an overview of different components of Headless Adaptive Forms. It also provides a list of all the supported components, their corresponding properties, and constraints.

Visual Studio Code Extension: [Visual Studio Code extension](#) to help create a valid JSON structure. It provides IntelliSense support and validation for JSON structure of forms along with common functions like add, delete, or rename components of a JSON structure.

Adaptive Forms version 2.0 specifications: Adaptive Forms version 2.0 specification provides a detailed information on all the components, constraints, and methods available to define Headless Adaptive Forms. The specification is available in [PDF](#) format.

HTTP and JavaScript APIs: [HTTP APIs](#) allow you to list, fetch, validate, submit, track submission status of headless forms. [JS APIs](#) helps you use Headless Adaptive Forms with any JavaScript based UI framework.

JSON Formula: It is an implementation of forms expression grammar to helps you query JSON structure and create rules for Headless Adaptive Forms. The grammar is a mashup of spreadsheet-like functions and operators and [JMESPath](#) a JSON query language. You can use the [playground](#) to explore JSON formula syntax and capabilities.

