



adaptTo()

EUROPE'S LEADING AEM DEVELOPER CONFERENCE

28th – 30th SEPTEMBER 2020

AMP-ify your online experience

with AEM and Core Components

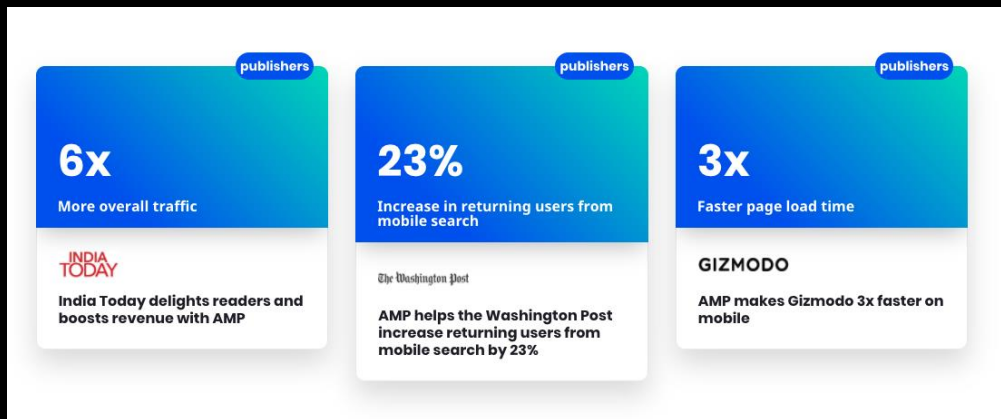
Burkhard Pauli - Adobe

What is AMP?

- Accelerated Mobile Pages
 - Web component framework
 - OpenJS Foundation, Google
 - Fast
 - Cross-platform

Why use AMP?

- Fast



- Better ranking

How does AMP work?

- Components
- Caches
- Static sizing
- Async Javascript, non-blocking
- Inline CSS, size-limited
 - Other: GPU-accelerated animations, prioritized resource loading (including fonts), preconnect API



- Added with help from Bounteous
- Since 2.11.0
- Custom **amp** selector
- Inline CSS
- Content-related AMP component JS
- [Demo](#)

Sample Content

Accordion with three items with each item being a layout container and containing sample content.

Item 1



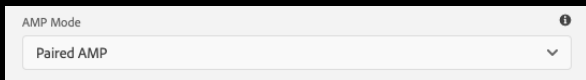
Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Eu mi bibendum neque egestas congue quisque egestas. Varius morbi enim nunc faucibus a pellentesque. Scelerisque eleifend donec pretium vulputate sapien nec sagittis.

Item 2

Item 3

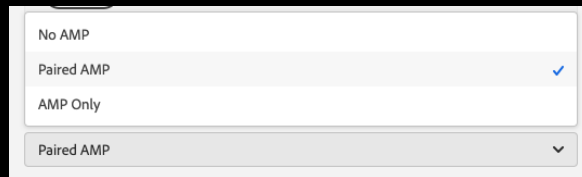
- Extension package & bundle
- Customized components
 - Accordion
 - Carousel
 - Form Container
 - Form Text Input
 - Image
 - PDF Viewer
 - Search
 - Sharing
 - Tabs

■ Content Policy



AMP Mode ?

Paired AMP



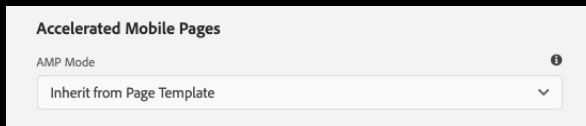
No AMP

Paired AMP

AMP Only

Paired AMP

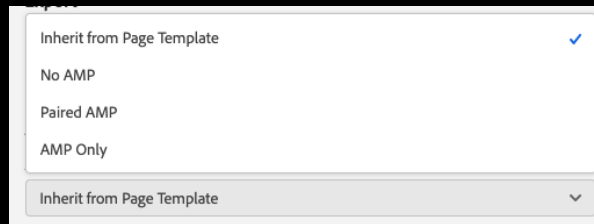
■ Page properties – Advanced tab



Accelerated Mobile Pages

AMP Mode ?

Inherit from Page Template



Inherit from Page Template

No AMP

Paired AMP

AMP Only

Inherit from Page Template

How to enable AMP with custom components

- Custom HTL script: `amp.html`
- Custom JS/CSS handling:
 - `customheadlibs.amp.html` – to include custom JS
 - `*.amp` clientlib category to inline CSS

- **Core Page (v2)**
 - **head.links.html**
 - `Set<String> Page#getComponentsResourceTypes ()`
- **AMP Page**
 - **ampheadlibs.html**
 - custom JS
 - inline CSS

- **Provides a list of paths**

```
List<String> ComponentFiles#getPaths()
```

- **Options**

- `resourceTypes` – collection of component resource types
- `filter` – regular expression
- `inherit` – look files up the component inheritance chain (default true)

■ Example

```
<sly data-sly-use.cf="`${'...ComponentFiles' @
    resourceTypes = page.componentsResourceTypes,
    filter='customheadlibs\\.amp\\.html'}"
data-sly-repeat.toInclude="`${cf.paths}"
data-sly-include="`${toInclude}"></sly>
```

- Provides a list of paths
 - Include/inline CSS and/or JS
- Options
 - `resourceTypes` – collection of component resource types
 - `filter` – regular expression
 - `inherit` – look files up the component inheritance chain (default true)
 - `categories` – collection of clientlib category

- **CSS Options**

- `media`

- **JS Options**

- `async`
- `defer`
- `crossorigin`
- `onload`

■ Example

```
<style amp-custom data-sly-use.cl="${'...ClientLibraries' @
    resourceTypes = page.componentsResourceTypes,
    filter='.*\\.amp'}">
    ${cl.cssInline @ context="unsafe"}
</style>
```

```
<sly data-sly-use.cl="${'...ClientLibraries' @
    categories='core.wcm.components.pdfviewer.v1',
    async=true}">${cl.jsIncludes @ context="unsafe"}</sly>
```



Thank you!