DD4T migration support

DXA's compatibility features facilitate a seamless migration from DD4T to SDL Digital Experience Accelerator 2.0.

Compatibility features

DXA includes the following compatibility features:

Incorporated latest DD4T version

DXA 2.0 incorporates the latest released versions of DD4T. For Java, this is version 2.1, while for .NET and Content Manager templates, the version is 2.2. For more information on the enhancements provided in these releases, refer to the DD4T website at http://dd4t.org.

In addition, with 2.0, DD4T is fully incorporated into the DXA distribution. You can drop all DD4T dependencies, migrate to DXA, and everything will continue to work.

Data model compatibility

For customers with existing implementations and content previously published to the DD4T data model, DXA provides the Model Service Converter. In addition, a DD4T Model Service Provider is available that works with Model Service. You can migrate to the DXA R2 data model according to a schedule that suits your needs.

Code compatibility

The DD4T code base is located within the DXA code base, which eliminates dependencies on DD4T code. Calls to the former DD4T namespace are automatically forwarded to DD4T in the DXA namespace, and DD4T web applications continue work as before.

Template compatibility

DXA 2.0 provides a new set of template building blocks (TBBs) designed for use with the R2 data model. The easier templating and extension points should minimize the migration effort.

If there is some reason that you still need to create legacy DD4T templates, DXA continues to provide the legacy TBBs that render legacy JSON for the DD4T data model.

Model Service Converter

The Model Service Converter supports customers with content published to the legacy DD4T format or with web applications that expect DD4T format.

The default published format for DXA version 2.0 and later is the R2 format, and R2 is expected in DXA 2.0 web applications. You might have, however, content was previously published to DD4T. Conversely, you might have existing modules for your 2.0 web applications that expect the DD4T format rather than the new R2 format. The Model Service Converter handles these conflicts.

The Model Service first determines whether the published format (either R2 or DD4T) matches the requested format. If there is a conflict between the published format and the requested format, then the converter converts the JSON to match the requested format. The following table summarizes the different scenarios.

Requested	Published	Actions taken by the Model Service and the Converter
R2	R2	The Model Service returns the R2 model to the R2 Content Provider. No conversion is necessary.
R2	DD4T	The Model Service Converter first converts the DD4T model to the R2 format and then returns the R2 model to the R2 Content Provider.
DD4T	DD4T	The Model Service returns the DD4T model to the DD4T Model Service Provider. No conversion is necessary.
DD4T	R2	The Model Service Converter first converts the R2 model to the DD4T format and then returns the DD4T model to the DD4T Model Service Provider.

DD4T Model Service Provider

DXA web applications load content abstractly through the Model Service rather than directly from the Content Data Store (the Broker database). The **DD4T Model Service Provider** is available for DD4T web applications so that they will behave in the same way as DXA applications, using the Model Service instead of making direct requests. A provider is available for both Java and .NET