

# Working with Templates

TS No-code Platform offers a wealth of functionality for working with Templates.

This can be used to setup auto-generated documents based on relational data in applications. Build advanced reports, contracts, bill of sales and other documents based on application input. Here is how it works.

- 1) Identify Variables and Sections in your document
- 2) Setup an application entity to reflect the variable inputs
- 3) Replace document variables with relevant Field Tokens
- 4) Insert tabular data in your document
- 5) Attach Document Template to the application entity

- [Creating a Document Template](#)
- [Setting up an Application Entity](#)
- [Testing the Template](#)

# Creating a Document Template

## Identify Variables and Sections

Create a Document Template in Microsoft Word or use an existing document. In the example below, we will “build” a simple contract based on an existing document. Go through the document to identify Variables and Sections. Variables could be anything like; name and contact details of the parties, effective date, duration, renegotiation terms, etc. Essentially anything that may vary from from one contract to the next.

## Field Tokens

Field Tokens are used to insert variable data. Field Tokens are always expressed using the syntax `#{SYSTEMNAME}`. Unfortunately, Microsoft Word tends to wrap elements in its own markup, which can destroy the integrity of the Field Token. To avoid this, we recommend that you turn off Track Changes for the document by navigating to the “Review” tab and switch Track Changes to “Off”. Copy Field Tokens into a notepad or text editor and convert everything to plain text. From here, copy and paste each Field Token in place in the Word document using the “Keep text only” paste function. This will preserve the integrity of your Field Tokens.

## Sections

A Document Section (Section) allows you to include or exclude entire sections of content in a document based on user input. A section is defined in the template document using markup similar to html. The syntax for sections is `<SYSTEMNAME> Section content </SYSTEMNAME>` and for alternative content `<SYSTEMNAME_ALT> Section content </SYSTEMNAME_ALT>`. In the following example we have defined a section called Terms, which will display either of two content pieces A or B depending on user input: `<TERMS> Section content A </TERMS>` or `<TERMS_ALT> Section content B </TERMS_ALT>`.

## Combine Sections and Field Tokens

To build advanced reports, complex contracts and other documents, you may need to combine the use of Sections and Field Tokens. You can add any number of Field Tokens within a Section, provided that you follow the correct syntax:

`<SECTION> Section Content A and #{DATE} variables </SECTION>` and alternatively `<SECTION_ALT> Section Content B and #{DATE} variables </SECTION_ALT>`.

# Tabular Data

When building financial reports, executive summaries and other business reports, you will often need to include tabular data (tables) in your template documents. This works in much the same way as Field Tokens and make use of the same syntax `${SYSTEMNAME}`. This allows you to include Sub Form Lists from the entity as variable data.

In order to control the visual appearance of tables in documents based on the template, you will need to add some additional configurations to the application entity. The table below shows a list of the existing configuration options:

Config Name	Description	Config Value (HEX for Word)
WordTableColorBorder	Define table border color	333333 (without # prefix)
WordTableColorCells	Define background color of cells	(no value = transparent)
WordTableColorHeader	Define background color of table header	333333 (without # prefix)

# Setting up an Application Entity



Once you have identified the variable data in the document you can begin building the Application Entity to feed into the document or use an existing entity. Every field in a TS Application has a unique system name (all caps) e.g. `DATE`, `USER`, `WEIGHT`, etc. Each field name can be used as a “Field Token” by replacing variable data with the relevant Field Token. Example: `${DATE}` will insert the field value of “DATE” in the document.

## Attach Template

When you have completed your template document with all the relevant Field Tokens and Sections markup you are ready to attach your template to the application entity.

**Field** remove manual edit assign translate siblings changes ⏻

---

Type	Basic: Text <span>⌵</span> 
Title	Name <span>NAME</span> <span>usage</span>
Configuration	<span>Edit configurations</span> 
Options	<input checked="" type="checkbox"/> Include field value in item Resume
Display	<input checked="" type="checkbox"/> Show by default in List views
	<input checked="" type="checkbox"/> Show by default in Lists of children on parent items
Validation	<input type="checkbox"/> Perform validation (optionally use onclick <span>javascripts</span> )
Block	Default <span>⌵</span>
Page	Show on all pages <span>⌵</span>

---

Submit Cancel

# Testing the Template

When you have finished the attachment process, go to the front-end to test the template. Navigate to the relevant entity, select an existing record or create a new record for testing. On the record, click the “Output” button in the toolbar and select the Template from the output dropdown list (see example below).



Click the desired output document. This generates an output file, in the specified format, based on the template and the variable data of that particular record. Depending on the selected output format configuration, the generated file will automatically open in your web browser or download to your desktop download folder.