INTERNATIONAL COMPUTATIONAL LAW JOURNAL

CODEX SOURCE ARCHIVE

Video Tutorial: How to Create a Simple Docassemble Interview

Sam Harden

Computational Law Report · Release 1.0 · 2019-12-06

Reader edition prepared by The Research University under CC BY 4.0. This cover was added; the original PDF pages follow unchanged. Original authorship, publication identity and source links are preserved. This independent republication is not sponsored or endorsed by Stanford University, Stanford Law School, CodeX or MIT.

Volume 1 · Issue 1 · Foundations: Law Becomes Computable

Original source and license evidence are recorded on the article landing page.

MIT Computational Law Report • Release 1.0

Video Tutorial: How to Create a Simple Docassemble Interview

Sam Harden

Published on: Dec 06, 2019

URL: https://law.mit.edu/pub/howtocreateasimpledocassembleinterview

License: Creative Commons Attribution 4.0 International License (CC-BY 4.0)

Video

Visit the web version of this article to view interactive content.

How to Create a Simple Docassemble Interview

Overview

Docassemble is a free, open-source expert system for guided interviews and document assembly, based on Python, YAML, and Markdown. Docassemble has strong implications for the potential of computational law because it is a free and straightforward tool for computing legal processes and legal functions.

This lesson provides an overview about how to create a simple Docassemble interview. Complete these steps and anyone can go online and fill out your very own Docassemble interview.

Note: Steps are organized based on the software that is used to perform each step. Just in case you lose your place and have to start over, we have included a link to the specific place in the video where each step takes place.

Microsoft Word (or similar software)

1. Using a traditional document from Microsoft Word, need to identify the variables that we want to account for in our docassemble interview. To create a variable that will be read in Docassemble, use curly braces for example: `{{ plaintiff_name }}`. Note: there cannot be spaces within a variable name. ***(Link)***

Note: If you do not want to use Microsoft Word, you may also use Pages on Mac or GoogleDocs, so long as you save the document in the .docx file format.

2. Once you have added all of the variables, we need to save the document as a template file. It is best practice here to save without any spaces. ***(Link)***

Docassemble

3. Under template section in Docassemble, click Browse, find your template (.docx file), and click upload. ***(Link)***

Note: To get to the template section in Docassemble, type in the URL for your AWS EC2 instance, click on your username in the upper right hand corner of the page for Docassemble, select Playground, click Folders --> Templates, and you will be on the right page.

4. Navigate back to the playground by clicking on your username, and playground. ***(Link)***

5. Start a new interview by clicking "+ Add" and then, under Example blocks, on the left side of the page, add the text for "Using a .docx template" and click insert. ***(Link)***
6. Be sure to make this mandatory. Docassemble requires a mandatory variable to complete the interview. ***(Link)***
7. Replace the .docx template file with the template that we have uploaded. Then click save. ***(Link)***
8. Now, all the variables that we haven't defined will show up under "undefined names." ***(Link)***
9. Create a set of questions that help answer the variables that you need for your document. ***(Link)***
10. If there's a possessive or a quote, you will need to add quotation marks around the question because Python requires quotes to be closed out. ***(Link)***
11. Once all variable names have been assigned, click "save." ***(Link)***
12. If you get no errors, click "save and run." ***(Link)***
13. After you fill out the information from your interview, you will see options to render the document as either a .pdf or .docx ***(Link)***
14. From here, in order to make the document usable outside of the Playground, go back to the playground and click "Folders" and then click "Packages" ***(Link)***
15. Name the package, and fill out the related information. ***(Link)***

Note: It is a best practice to add information about the README, to help people who want to access this file understand what is in it.

16. After you save your interview in Docassemble, you will need to Install it. This updates all of the packages and installs the interview packages so that the file can be used by people outside of your Docassemble instance. ***(Link)***
17. Click Return. ***(Link)***
18. The docassemble interivew will then generate a URL. A helpful tip is to name the package the same as the YAML file. ***(Link)***

Bonus

Note: the formula for the URL is `[name of your ec2 domain]` + `"interview?i=docassembly."` + `[package name]` + `":data/questions/"` + `[yaml file name]`

19. For ease of use, it is best to get rid of questions from Example Blocks that are not needed. ***(Link)***
20. For variables that need to be setup with rails, it is helpful to list a series of choices or datatypes (e.g., integers, choices) ***(Link)***
21. Each time you complete a new interview, increment to the latest version of the document. ***(Link)***

Want to learn more about what Sam Harden is up to?

Follow his latest work on his youtube channel, LegalTech School or check out his website samharden.github.io