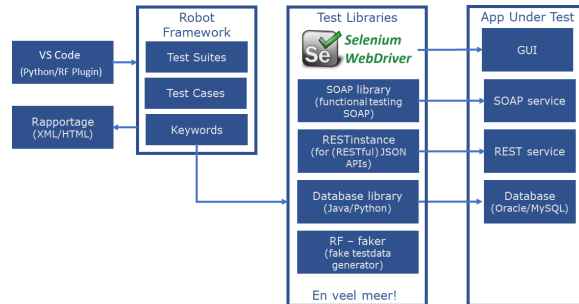


Robot Framework

Robot Framework is een generiek testautomatisering framework ontwikkelt voor end-to-end acceptatie testen en het borgen van acceptatiecriteria op basis van (behavior/acceptance) test driven development. Een aantal eigenschappen van het framework (WIP):



- **Open source**

Robot Framework is ontstaan binnen Nokia Networks en is sinds 2008 open source (uitgebracht onder Apache 2.0 licentie). De Robot Framework community leeft sindsdien vooral op Github en er zijn nog zeer actieve discussiegroepen op Google Groups. De ondersteunende kanalen kan je [hier](#) vinden. Toegang tot het community Slack channel kan [hier](#) worden aangevraagd.

- **Modulaire architectuur**

Robot Framework is geschreven in Python, en draait ook op Jython (JVM) en IronPython (.NET). Dat maakt het OS en applicatie onafhankelijk. Aan de rechterkant zie je de structuur van Robot Framework. Door de modulaire opbouw kan het makkelijk uitgebreid worden met bestaande of nieuwe testlibraries.

- **Keyword gedreven**

Keywords zijn eenvoudig leesbare combinaties van woorden. Het is mogelijk om keywords van verschillende testlibraries te gebruiken of zelf op verschillende niveaus te definiëren. Door middel van keywords kan een gewenste structuur worden aangebracht en stukken code binnen elke willekeurige test hergebruikt worden.

Dit vergroot de toegankelijkheid voor zowel testers met en zonder technische kennis. Testers met technische kennis kunnen keywords op zo laag niveau als nodig definiëren, zodat testers met beperkte technische kennis deze kunnen gebruiken bij het samenstellen van de testen.

Om de testen onderhoudbaar te houden is het van groot belang dat er een eenduidige structuur en werkwijze wordt aangehouden. Deze structuur wordt [hier](#) beschreven

- **Libraries**

De test libraries die door Robot Framework worden ontsloten (mbv keywords) leveren de daadwerkelijke functionaliteiten om je testen te kunnen automatiseren. Functionaliteiten die het mogelijk maken om met de te testen applicatie te interacteren. Er zijn verschillende libraries die op verschillende lagen met de applicatie kunnen interacteren, een paar voorbeelden kan je in het plaatje rechts zien. Zie [hier](#) de verschillende libraries die beschikbaar zijn. Let op dat hier libraries tussen zitten die geschreven zijn in zowel Python als Java. Het is mogelijk om verschillende libraries tegelijk te gebruiken, deze moeten dan allemaal wel in dezelfde taal geschreven zijn. Libraries geschreven in Python hebben daarom altijd de voorkeur. Door recente ontwikkelingen kan er ook een 'brug' opgezet worden zodat er vanuit Python gebruik gemaakt kan worden van Java gebaseerde libraries zoals de Browser library en Playwright.

Ook is het mogelijk je eigen library te maken. Dit kan je het beste in Python schrijven, maar het is ook mogelijk om dit in een andere taal te schrijven. Dan kan je deze echter niet in combinatie met andere libraries te gebruiken.

Voor een lijst met beschikbare eigen gemaakte packages binnen UWV klik [hier](#).

- **Makkelijk te integreren**

Door de modulaire opbouw is het makkelijk om Robot Framework met andere tooling te integreren. De robot testsuites kunnen in versiebeheer gecentraliseerd en beheerd worden. Zie [hier](#) voor uitleg over het plaatsen van testautomatisering in versiebeheer.

Het is mogelijk de robot tests uit te voeren via de Command Line Interface (CLI), en de output van de tests worden in XML format gegenereerd. Hierdoor is integratie van Robot Framework met CI en build tools (Plugin, Ant, Maven) mogelijk. Meer informatie over testen via de CLI vind je [hier](#).