IT4KT

Kód ITMS projektu: 26220220123

Agentúra Ministerstva školstva, vedy, výskumu a športu SR pre štrukturálne fondy EÚ

Návod: E2 - Structure Checker

Riešiteľský kolektív	Miroslav Biňas

História dokumentu:

Verzia	Autor(i)	Dátum	Sumár zmien
01	Miroslav Biňas	Október 2012	Prvá verzia dokumentu
02	Miroslav Biňas	Február 2013	Príklad použitia konzolového klienta
03	Miroslav Biňas	Jún 2014	Aktualizácia motivácie
04	Miroslav Biňas	Júl 2014	Aktualizácia použitia služby
05	Miroslav Biňas	August 2014	Aktualizácia požiadaviek na inštaláciu, použitie služby, rozšírenie príkladov použitia, motivácia

Motivácia

Študentské zadania sú častokrát kvôli svojej komplexnosti reprezentované formou archívov, ktoré umožňujú viacero súborov odovzdať ako jeden (napr. `zip`, `rar`). Učiteľ pri ich preberaní najprv musí skontrolovať obsah takéhoto archívu, či obsahuje všetky požadované súbory. Táto činnosť môže byť pomerne zdĺhavá, pretože pomenovávanie priečinkov a súborov v odovzdávanom balíčku môže byť u každého študenta rôzna a nájsť potrebný súbor môže zabrať istý čas.

Motiváciou pre vznik služby *Structure Checker* je práve automatizácia tohto procesu. Služba zabezpečí kontrolu existencie každého požadovaného súboru a v špeciálnych prípadoch vie skontrolovať aj obsah súborov na prítomnosť špecifického textu (napr. študentove iniciály alebo kontrola existencie požadovaných funkcií/metód pri programátorských zadaniach).

Učiteľ najprv vytvorí všeobecný opis obsahu balíčku, ktorý majú študenti odovzdávať v podobe XML súboru. Jeho "ľudskú" (teda čitateľnú) podobu zverejní aj študentom, aby títo vedeli, ako má štruktúra balíčku vyzerať. V procese preberania zadania potom učiteľ spustí nástroj *Structure Checker* s týmto opisným XML súborom a študentovým zadaním. Výsledkom kontroly bude správa v požadovanom tvare (XML, JSON, text), na základe ktorej sa je možné dozvedieť, na čo študent pri zostavovaní balíčku zabudol. Vhodným spôsobom je možné tento nástroj zreťaziť do komplexnejšieho postupu kontroly zadaní (napr. pomocou webovej služby).

Inštalácia

Pre nainštalovanie služby odporúčame vytvoriť vlastné prostredie pomocou nástroja `virtualenv` a do neho nainštalovať balíčky nachádzajúce sa v súbore `requirements.txt`. Postup inštalácie by teda mohol vyzerať nasledovne:

```
$ virtualenv env
$ . env/bin/activate
$ (env) pip install -r requirements.txt
```

Následne je potrebné nakonfigurovať webový server *Apache*. V prípade, že používate *Ubuntu Server*, otvorte súbor `/etc/apache2/sites-enabled/000-default` a vložte do neho nasledujúce riadky:

```
WSGIDaemonProcess checker user=tester group=tester threads=5
WSGIScriptAlias /checker /path/to/service.wsgi
WSGIScriptReloading On

<Directory /path/to/service>
   WSGIProcessGroup checker
   WSGIApplicationGroup %{GLOBAL}
```

```
   Order deny,allow
   Allow from all
</Directory>
```

Pokiaľ používate inú distribúciu, upravte pre ňu potrebný súbor.
Po úprave už stačí len webový server reštartovať príkazom:

```
$ sudo service apache restart
```

Použitie služby

Webovú službu je možné integrovať do vlastných riešení a vytvárať tak kompozitné webové služby (webové služby pozostávajúce z viacerých služieb). Použitie nie je viazané na žiadnu technológiu a príklad použitia priamo z príkazového riadku pomocou nástroja `curl` demonštruje nasledujúci riadok:

```
$ curl -X POST http://url:port/ -F template=@template.xml -F \
package=@package.zip
```

kde:

- `url:port` reprezentuje lokáciu služby
- `template.xml` predstavuje XML súbor pre kontrolu balíčka; tento parameter je povinný
- `package.zip` predstavuje umiestnenie zip balíčka, ktorý má byť skontrolovaný

Pomocou atribútu `Accept` v hlavičke HTTP protokolu je možné požiadať aj o konkrétny formát výstupu. Služba aktuálne podporuje nasledujúce formáty výstupu: XML, JSON, HTML a textový výstup, napr.:

```
$ curl -H "Accept: application/json" -X POST http://url:port/ \
-F template=@template.xml -F package=@package.zip
```

Súbor pre opis odovzdávaného balíčka

Názvy súborov a priečinkov sú citlivé na veľkosť písmen (case sensitive).

Element `PACKAGE`

atribút	povinný	hodnoty	opis
`name`	Y	`string`	Tvar názvu odovzdávaného balíčka reprezentovaný pomocou regulárneho výrazu. Pomocou tohto atribútu bude kontrolovaný názov odovzdávaného balíčka.

Element `DIRECTORY`

Tento element opisuje priečinok, ktorý sa v balíčku môže nachádzať.

atribút	povinný	hodnoty	opis
`pattern`	N	`string`	Regulárny výraz, pomocou ktorého sa bude kontrolovať názov priečinku. Ak tento nie je zadaný, bude sa názov testovať pomocou hodnoty atribútu `name`.
`name`	Y	`string`	Tvar názvu priečinku. Názov je v tvare “čitateľnom” pre človeka, pretože sa použije v chybovej správe alebo varovaní, pokiaľ tento priečinok bude chýbať. Rovnako sa použije pre kontrolu názvu priečinka, ak bude chýbať atribút `pattern`.
`mandatory`	N	`true/ false`	Atribút udáva, či sa priečinok musí v balíčku nachádzať alebo nie. Ak nie je povinný a priečinok v balíčku nebude umiestnený, výsledkom bude varovanie. Ak bude nastavený ako povinný a priečinok nebude v balíčku umiestnený, výsledkom bude chyba. Predvolená hodnota je `true`.
`casesensitive`	N	`true/ false`	Určuje, či sa majú pri kontrole názvu rozlišovať veľké a malé písmená. Predvolená hodnota je `true`.

Element `FILE`

Tento element opisuje súbor, ktorý sa v balíčku môže nachádzať.

atribút	povinný	hodnoty	opis
`pattern`	N	`string`	Regulárny výraz, pomocou ktorého sa bude kontrolovať názov súboru. Ak tento nie je zadaný, bude sa názov testovať pomocou hodnoty atribútu `name`.
`name`	Y	`string`	Tvar názvu súboru. Názov je v tvare “čitateľnom” pre človeka, pretože sa použije v chybovej správe alebo varovaní, pokiaľ tento súbor bude chýbať. Rovnako sa použije pre kontrolu názvu súboru, ak bude chýbať atribút `pattern`.
`mandatory`	N	`true/ false`	Atribút udáva, či sa súbor musí v balíčku nachádzať alebo nie. Ak nie je povinný a súbor v balíčku nebude umiestnený, výsledkom bude varovanie. Ak bude nastavený ako povinný a súbor nebude v balíčku umiestnený, výsledkom bude chyba. Predvolená hodnota je `true`.

casesensitive	N	true/false	Určuje, či sa majú pri kontrole názvu rozlišovať veľké a malé písmená. Predvolená hodnota je `false`.

Subelement CONTAINS

Pomocou tohto elementu je možné overiť, či sa v danom súbore nachádza potrebný reťazec. Tento je možné zadať pomocou vhodného regulárneho výrazu. V prípade potreby je možné rozlišovať veľké a malé písmená pomocou regulárneho výrazu.

atribút	povinný	hodnoty	opis
pattern	Y	string	Regulárny výraz reprezentujúci výraz, ktorý sa má nachádzať v uvedenom súbore.
message	Y	string	Text chybovej správy, ktorá sa zobrazí, ak uvedený `pattern` nebol v súbore nájdený.
line	N	integer	Číslo riadku, na ktorom sa bude `pattern` kontrolovať.

Príklady použitia

Príklad č. 1

Študenti majú odovzdať zabalený balíček v nasledovnej štruktúre:

```
|--- zadanie.txt
+--- main.c
```

Súbor `zadanie.txt` je textový a obsahuje znenie zadania.
Súbor `main.c` obsahuje riešenie študenta, pričom sa v ňom musí nachádzať riadok nasledujúcej syntaxe:
`// Vypracoval: Meno Priezvisko, 2012`
Názov balíčka nech je v tvare `zadanie-z-programovania.zip`

XML súbor s opisujúci takýto balíček môže vyzerať nasledovne:

```
<?xml version="1.0" encoding="UTF-8"?>
<package xmlns="http://kpi.fei.tuke.sk/IT4KT"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="http://kpi.fei.tuke.sk/IT4KTschema.xsd"
   name="zadanie-z-programovania.zip">

   <file name="main.c">
```

```
      <contains pattern=".*// Vypracoval: [A-Z][a-z]+ [A-Z][a-z]+,
2012" message="Chyba ti hlavicka v zadanom formate."/>
   </file>

   <file name="zadanie.txt"/>
</package>
```

Príklad č. 2

Študenti majú odovzdať zabalený balíček v nasledovnej štruktúre:

```
.
|-- ps1
|   |-- tesco.c
|   `-- tiktak.c
`-- README
```

Jednotlivé súbory majú nasledujúci význam:

- `/ps1/tiktak.c` - Zdrojový kód riešenia úlohy s názvom *Tik Tak*.
- `/ps1/tesco.c` - Zdrojový kód riešenia úlohy s názvom *Samoobslužná pokladňa*.
- `/README` - Súbor, v ktorom budú uvedené informácie o autorovi.

XML súbor s opisujúci takýto balíček môže vyzerať nasledovne:

```
<?xml version="1.0" encoding="UTF-8"?>
<package xmlns="http://kpi.fei.tuke.sk/IT4KT"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://kpi.fei.tuke.sk/IT4KTschema.xsd"
        name="prog-2014(-[a-z][a-z0-9])?\.zip">

    <directory name="ps1">
        <file name="tiktak.c"/>
        <file name="tesco.c"/>
    </directory>

    <file name="README"/>
</package>
```

Príklad č. 3

Tento príklad je vlastne modifikáciou prikladu č. 2. Rozdiel bude akurát v podobe obsahu súboru `README`.

Študenti majú odovzdať zabalený balíček v nasledovnej štruktúre:

```
.
|-- ps1
|   |-- tesco.c
|   `-- tiktak.c
`-- README
```

Jednotlivé súbory majú nasledujúci význam:

- `/ps1/tiktak.c` - Zdrojový kód riešenia úlohy s názvom *Tik Tak*.
- `/ps1/tesco.c` - Zdrojový kód riešenia úlohy s názvom *Samoobslužná pokladňa*.
- `/README` - Súbor, v ktorom budú uvedené informácie o autorovi v nasledujúcej podobe:

```
NAME        : NAME SURNAME
STUDENT ID  : ab123cd
E-MAIL      : name@student.tuke.sk
YEAR        : 9999
GROUP       : A1
TOKEN       : kduhrkISHHdjFUD8767h
```

XML súbor s opisujúci takýto balíček bude vyzerať podobne, ako v prípade príkladu č. 2. Rozdiel bude akurát v kontrole štruktúry súboru `README`:

```
<?xml version="1.0" encoding="UTF-8"?>
<package xmlns="http://kpi.fei.tuke.sk/IT4KT"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://kpi.fei.tuke.sk/IT4KTschema.xsd"
        name="prog-2014(-[a-z][a-z0-9])?\.zip">

    <directory name="ps1">
        <file name="tiktak.c"/>
        <file name="tesco.c"/>
    </directory>

    <file name="README">
        <contains pattern="^NAME\s*:\s*[A-Z]+( [A-Z]+)+$"
message="Name and surname are missing or have wrong format."/>
        <contains pattern="^STUDENT
ID\s*:\s*[a-z]{2}[0-9]{3}[a-z]{2}$" message="Student ID is missing or
has wrong format."/>
        <contains
pattern="^E-MAIL\s*:\s*([a-z0-9A-Z_-]+)(\.[a-z0-9A-Z_-]+)*@[a-zA-Z0-9
_-]+(\.[a-zA-Z0-9_-]+)*(\.[a-zA-Z]{2,4})" message="E-mail address is
missing or has wrong format."/>
        <contains pattern="^YEAR\s*:\s*[0-9]{4}$" message="Year is
missing or has wrong format."/>
        <contains pattern="^GROUP\s*:\s*[a-zA-Z][a-zA-Z0-9]$"
```

```
message="Group is missing or has wrong format."/>
        <contains pattern="^TOKEN\s*:\s*[a-zA-Z0-9]+$" message="Token
is missing or has wrong format."/>
    </file>
</package>
```