

recordschool

sous forme de services web

Introduction

Ecole-chercheurs, modèle recordschool

L'école-chercheurs « **Approches interdisciplinaires de la modélisation des agroécosystèmes** » organisée par l'[INRA](#) en mars 2017 à Sainte Foy Lès Lyon s'appuie sur un modèle de la plateforme [RECORD](#) . Ce modèle correspond au dépôt de modèles **recordschool** des services web [erecord](#) .

Services web erecord

Les services web **erecord** permettent de piloter à distance la simulation de modèles de la plateforme [RECORD](#) .

Accès aux services web erecord : <http://erecord.toulouse.inra.fr> .

Pour en savoir plus : voir le document [Présentation générale des services web erecord](#) .

Services web erecord pour recordschool

Ce document présente un certain nombre d'appels du modèle recordschool sous forme de services web erecord.

Les exemples donnés illustrent la variété des formes d'appel des services web erecord : appel sous l'interface web utilisateur erecord, une interface en ligne de commande, un programme informatique...

A. Identifier son simulateur

Chaque **simulateur** présent dans les services web erecord possède un identifiant **Id** qui sert à le désigner dans les requêtes erecord à son sujet.

On va s'intéresser ici au simulateur record_school.vpz du modèle RS_FilRouge de l'Ecole Chercheur RECORD.

Le simulateur **record_school.vpz** a pour **Id=764**.

Pour aller plus loin

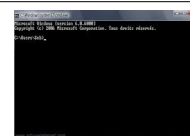
Pour savoir comment trouver l'Id d'un simulateur : voir « **Annexe A. Connaître les simulateurs disponibles** » dans le document [Annexes](#).

B. Observer un simulateur

Les services web erecord permettent d'obtenir les conditions de simulation d'un simulateur : son paramétrage, le début et la durée de simulation.



interface en ligne de commande



On peut demander ces informations depuis une interface en ligne de commande.

Le paramètre de requête **vpz** permet de sélectionner le simulateur (**record_school.vpz**). Pour ne garder que les paramètres agronomiques qui nous intéressent, on utilise le paramètre de requête **parselect** qui permet de filtrer les paramètres retournés.

Requête depuis une interface en ligne de commande

```
> curl -H "Content-Type: application/x-www-form-urlencoded" -d  
'vpz=764&style=compactlist&format=json&parselect=cond_DE_City&parselect=cond_DE_ModOstral_Farm1.TotalNbEwes&parselect=cond_DE_ModOstral_Farm2.TotalNbEwes&parselect=cond_DE_ModOstral_Farm3.TotalNbEwes&parselect=cond_DE_ModOstral_Farm4.TotalNbEwes&parselect=cond_DE_ModOstral_Farm5.TotalNbEwes'  
http://erecord.toulouse.inra.fr:8000/vpz/input/
```

Réponse

```
{"begin":2454024.0,"duration":5365.0,"cond_DE_ModOstral_Farm5.TotalNbEwes":"[50.0]","cond_DE_ModOstral_Farm4.TotalNbEwes":"[1000.0]","cond_DE_ModOstral_Farm3.TotalNbEwes":"[0.0]","cond_DE_ModOstral_Farm2.TotalNbEwes":"[200.0]","cond_DE_City.Inhabitants":"[10000.0]","cond_DE_City.YearlyUse":"[100.0]","cond_DE_ModOstral_Farm1.TotalNbEwes":"[0.0]","views":[],"outs":[],"conds":["cond_DE_City","cond_DE_ModOstral_Farm1","cond_DE_ModOstral_Farm2","cond_DE_ModOstral_Farm3","cond_DE_ModOstral_Farm4","cond_DE_ModOstral_Farm5"]}
```

Pour aller plus loin

Voir « **Annexe B. Observer un simulateur** » dans le document [Annexes](#).

C. Simuler

Les services web erecord permettent de modifier les conditions de simulation d'un simulateur et d'obtenir ses résultats de simulation.



navigateur web



On peut construire une requête de simulation dans un navigateur web.

Le paramètre de requête **format=html** permet alors d'obtenir la réponse dans une page html sur laquelle les résultats de simulation peuvent être visualisés dans un tableau et sous forme graphique. Le paramètre de requête **vpz** permet de sélectionner le simulateur (**record_school.vpz**). Le paramètre de requête **outselect** permet de filtrer les données de sortie retournées.

Construction de la *requête* dans un navigateur web

> saisie de l'**URL** de la requête <http://erecord.toulouse.inra.fr:8000/vpz/output/> dans la barre d'adresse du navigateur.

> sélection de l'**onglet 'Raw data'**

> sélection de '**application/x-www-form-urlencoded**' pour le champ '**Media type**',

> dans le champ '**Content**', saisie des paramètres de requête suivants :

`vpz=764&style=compact&format=html&plan=single&restype=dataframe&outselect=common`

> bouton '**POST**' pour lancer la requête de simulation.

erecord - Mozilla Firefox

erecord

erecord.toulouse.inra.fr:8000/vpz/output/

Django REST framework v 3.1.0

Raw data HTML form

Media type: application/x-www-form-urlencoded

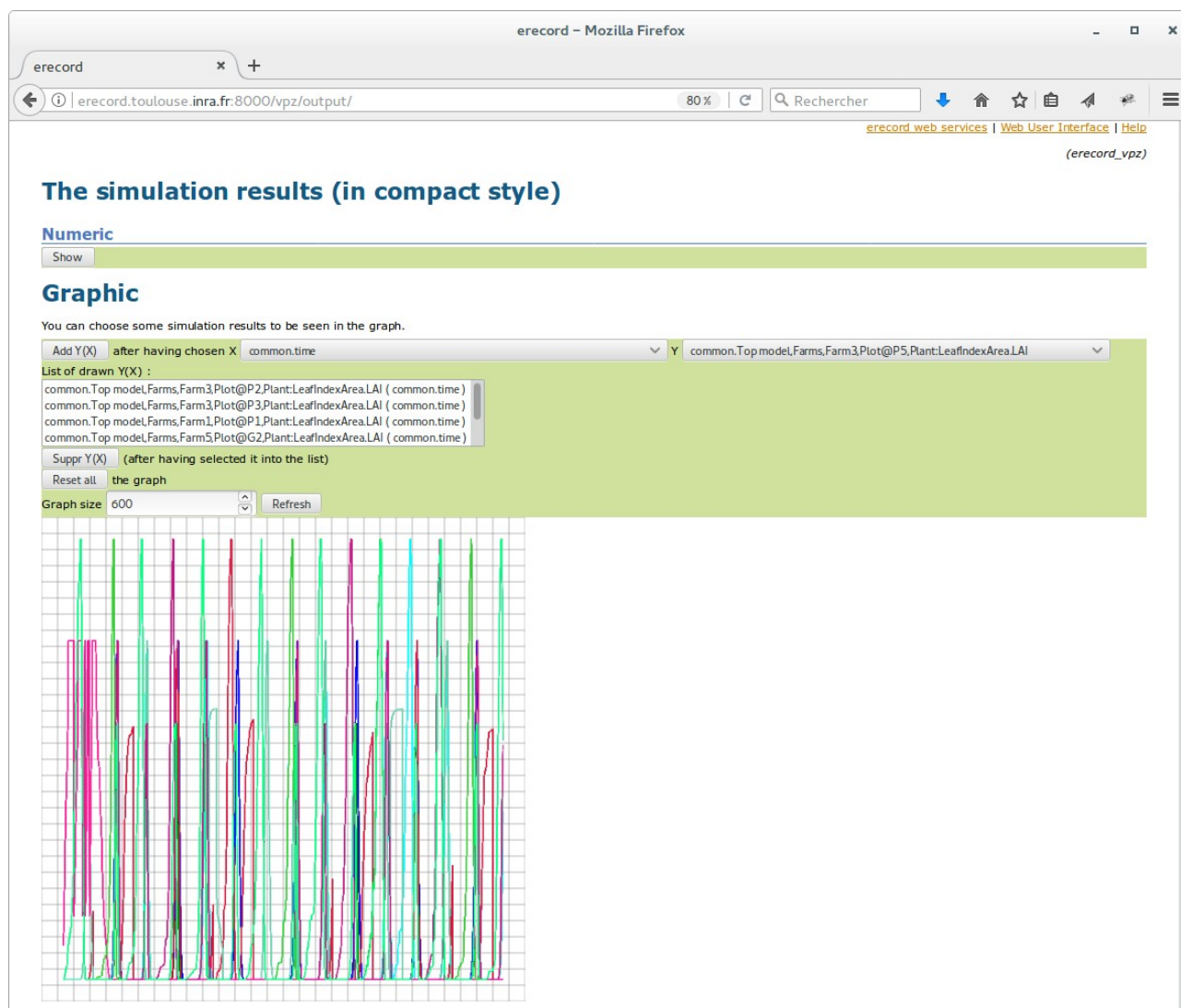
Content: vpz=764&style=compact&format=html&plan=single&restype=dataframe&outselect=common

POST

Réponse

On obtient les résultats de simulation du simulateur **record_school.vpz** :

- Résultats numériques (bouton Numeric / « **Show** »).
- Résultats graphiques (bouton « **Add Y(X)** » après avoir choisi « **X** » et « **Y** »).



Pour aller plus loin

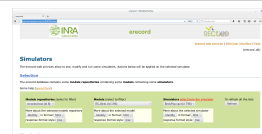
Voir « **Annexe C. Simuler** » dans le document [Annexes](#).

D. Travailler par fichier xls

Les services web erecord permettent de travailler par fichier xls et d'obtenir ainsi, regroupés dans un même **fichier xls**, les conditions de simulation et les résultats de simulation d'un simulateur.



interface utilisateur erecord



On peut travailler par fichier xls sous l'interface web utilisateur erecord <http://erecord.toulouse.inra.fr:8000/home>, en enchaînant les instructions de la rubrique « **Modify and run the simulator** », après avoir sélectionné le simulateur **recordschool.vpz (Id=764)** :

Requête

> Suivi des instructions « **First step : download the xls file to fill in** » qui permettent de recevoir le fichier xls contenant l'état initial du simulateur (paramètres...).

Réponse

Le fichier xls reçu est sauvegardé localement : [experiment_RS_FilRouge_initial.xls](#)

Opérations locales

Après avoir suivi les instructions « **Second step : build your own experiment plan** » permettant de modifier les conditions de simulation (paramétrage...), le fichier xls modifié est sauvegardé localement : [experiment_RS_FilRouge_modifie.xls](#)

Requête

> Suivi des instructions « **Third step : send your own experiment plan** » qui permettent de lancer la simulation basée sur le fichier xls modifié ([experiment_RS_FilRouge_modifie.xls](#)) et de recevoir le fichier xls contenant à la fois les résultats de simulation et les conditions de simulation.

Réponse

Le fichier xls reçu est sauvegardé localement : [experiment_RS_FilRouge_simu.xls](#).

Pour aller plus loin

Voir « **Annexe D. Travailler par fichier xls** » dans le document [Annexes](#).

E. Utiliser les services web dans un programme

Les services web erecord peuvent être appelés depuis un programme informatique écrit dans un langage de programmation supportant le protocole HTTP (Python, R, C++, Java, Php, etc), ce qui permet d'envisager divers développements d'applications basées sur des modèles accessibles sous les services web erecord.



programme informatique

Code
R

Exemple d'un programme R exécutant une simulation et en exploitant les résultats.

Simulation par appel des services web erecord

```
library('RCurl')
library('rjson')

header = c('Content-Type'='application/json', Accept='application/json')

postfields = toJSON(list(vpz=764,
  # duration=1000,
  # cond_DE_River.WatershedArea=800.0,
  # cond_DE_Dam.MaximumVolume=10000.0,
  style="compact", plan="linear", restype="dataframe",
  outselect="common",
  format="json"))

res = postForm(uri="http://erecord.toulouse.inra.fr:8000/vpz/output",
  .opts=list(postfields=postfields, httpheader=header))

responsedata = fromJSON(res)
res = fromJSON(responsedata$res)
```

Exploitation des résultats de simulation retournés par les services web erecord

```
time = res[[1]]$`common.time`
farm1_P4_LAI= res[[1]]$`common.Top model,Farms,Farm1,Plot@P4,Plant:LeafIndexArea.LAI`
farm1_P3_LAI= res[[1]]$`common.Top model,Farms,Farm1,Plot@P3,Plant:LeafIndexArea.LAI`
farm1_P1_LAI= res[[1]]$`common.Top model,Farms,Farm1,Plot@P1,Plant:LeafIndexArea.LAI`
farm1_P5_LAI= res[[1]]$`common.Top model,Farms,Farm1,Plot@P5,Plant:LeafIndexArea.LAI`
farm2_G5_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@G5,Plant:LeafIndexArea.LAI`
farm2_G2_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@G2,Plant:LeafIndexArea.LAI`
farm2_G7_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@G7,Plant:LeafIndexArea.LAI`
farm2_G6_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@G6,Plant:LeafIndexArea.LAI`
```

```
farm2_G1_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@G1,Plant:LeafIndexArea.LAI`  
farm2_P2_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@P2,Plant:LeafIndexArea.LAI`  
farm2_G3_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@G3,Plant:LeafIndexArea.LAI`  
farm2_G4_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@G4,Plant:LeafIndexArea.LAI`  
farm2_G8_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@G8,Plant:LeafIndexArea.LAI`  
farm2_P1_LAI= res[[1]]$`common.Top model,Farms,Farm2,Plot@P1,Plant:LeafIndexArea.LAI`  
farm3_P6_LAI= res[[1]]$`common.Top model,Farms,Farm3,Plot@P6,Plant:LeafIndexArea.LAI`  
farm3_P1_LAI= res[[1]]$`common.Top model,Farms,Farm3,Plot@P1,Plant:LeafIndexArea.LAI`  
farm3_P5_LAI= res[[1]]$`common.Top model,Farms,Farm3,Plot@P5,Plant:LeafIndexArea.LAI`  
farm3_P3_LAI= res[[1]]$`common.Top model,Farms,Farm3,Plot@P3,Plant:LeafIndexArea.LAI`  
farm3_P2_LAI= res[[1]]$`common.Top model,Farms,Farm3,Plot@P2,Plant:LeafIndexArea.LAI`  
farm3_P4_LAI= res[[1]]$`common.Top model,Farms,Farm3,Plot@P4,Plant:LeafIndexArea.LAI`  
farm4_G1_LAI= res[[1]]$`common.Top model,Farms,Farm4,Plot@G1,Plant:LeafIndexArea.LAI`  
farm5_P1_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P1,Plant:LeafIndexArea.LAI`  
farm5_G4_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@G4,Plant:LeafIndexArea.LAI`  
farm5_P3_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P3,Plant:LeafIndexArea.LAI`  
farm5_G3_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@G3,Plant:LeafIndexArea.LAI`  
farm5_P8_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P8,Plant:LeafIndexArea.LAI`  
farm5_P4_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P4,Plant:LeafIndexArea.LAI`  
farm5_P2_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P2,Plant:LeafIndexArea.LAI`  
farm5_G2_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@G2,Plant:LeafIndexArea.LAI`  
farm5_P10_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P10,Plant:LeafIndexArea.LAI`  
farm5_G5_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@G5,Plant:LeafIndexArea.LAI`  
farm5_G1_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@G1,Plant:LeafIndexArea.LAI`  
farm5_P7_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P7,Plant:LeafIndexArea.LAI`  
farm5_P5_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P5,Plant:LeafIndexArea.LAI`  
farm5_P9_LAI= res[[1]]$`common.Top model,Farms,Farm5,Plot@P9,Plant:LeafIndexArea.LAI`
```

```
plot(time, farm2_P1_LAI, type='l', col='magenta');  
lines(time, farm2_G5_LAI, col='green'); lines(time, farm2_G2_LAI, col='red');  
lines(time, farm2_G7_LAI, col='blue'); lines(time, farm2_G6_LAI, col='gray');  
lines(time, farm2_G1_LAI, col='black'); lines(time, farm2_P2_LAI, col='brown');  
lines(time, farm2_G3_LAI, col='orange'); lines(time, farm2_G4_LAI, col='pink');  
lines(time, farm2_G8_LAI, col='cyan');
```

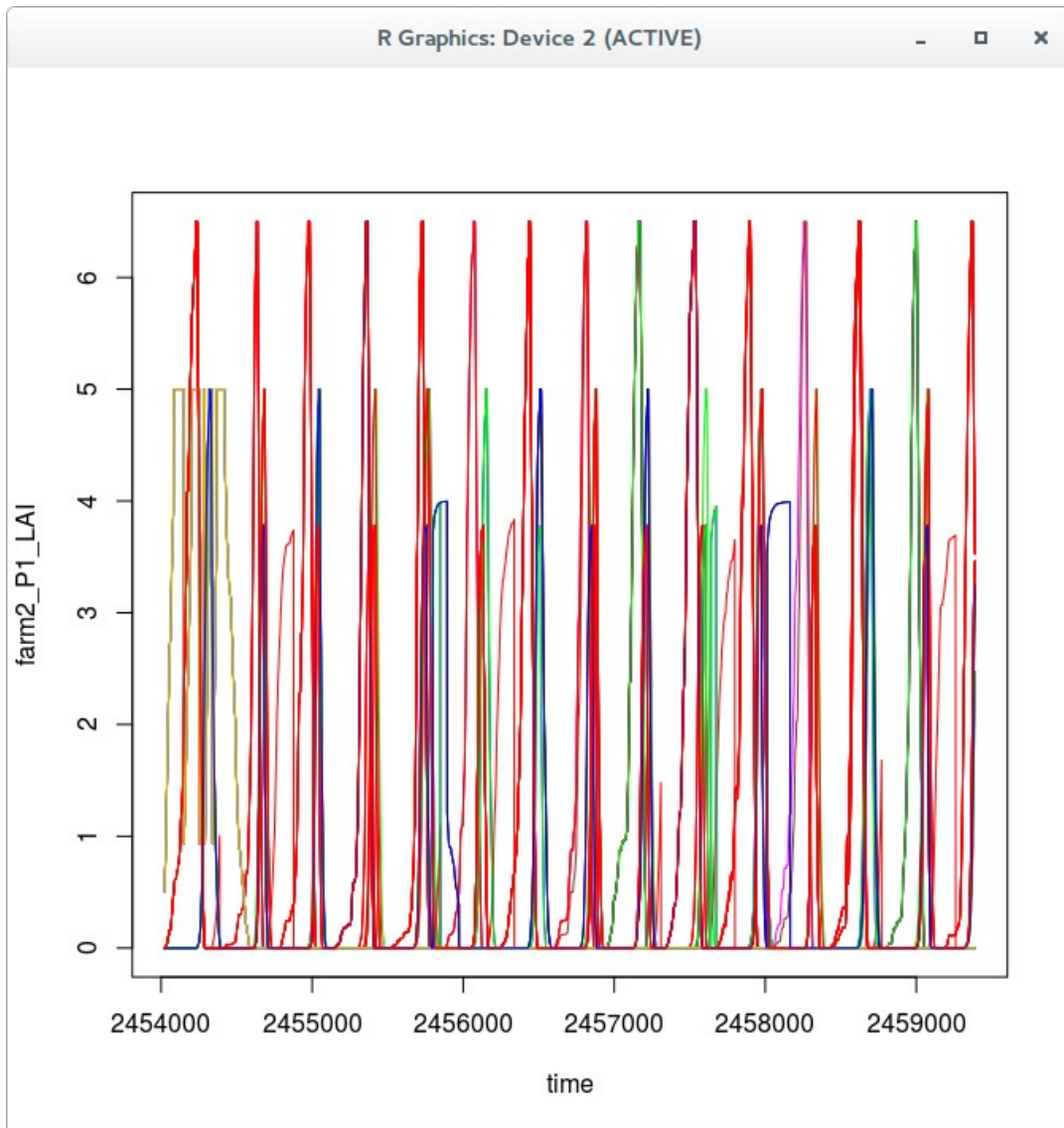
```
lines(time, farm1_P4_LAI, col='red'); lines(time, farm1_P3_LAI, col='red');  
lines(time, farm1_P1_LAI, col='green'); lines(time, farm1_P5_LAI, col='red');
```

```
lines(time, farm3_P6_LAI, col='red'); lines(time, farm3_P1_LAI, col='red');  
lines(time, farm3_P5_LAI, col='red'); lines(time, farm3_P3_LAI, col='red');  
lines(time, farm3_P2_LAI, col='red'); lines(time, farm3_P4_LAI, col='red');
```

```
lines(time, farm4_G1_LAI, col='green');
```

```
lines(time, farm5_P1_LAI, col='blue'); lines(time, farm5_G4_LAI, col='gray');  
lines(time, farm5_P3_LAI, col='orange'); lines(time, farm5_G3_LAI, col='magenta');  
lines(time, farm5_P8_LAI, col='red'); lines(time, farm5_P4_LAI, col='black');  
lines(time, farm5_P2_LAI, col='green'); lines(time, farm5_G2_LAI, col='brown');  
lines(time, farm5_P10_LAI, col='red'); lines(time, farm5_G5_LAI, col='cyan');  
lines(time, farm5_G1_LAI, col='orange'); lines(time, farm5_P7_LAI, col='red');  
lines(time, farm5_P5_LAI, col='blue'); lines(time, farm5_P9_LAI, col='red');
```


La représentation graphique obtenue



ANNEXES

Annexe A. Connaitre les simulateurs disponibles

Les modèles de la plateforme **RECORD** qui sont présents sous erecord sont rangés dans des dépôts de modèles (« models repositories »). Chaque modèle (« model ») contient un certain nombre de simulateurs (« simulators »).

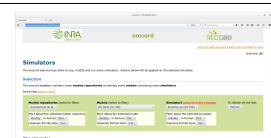
Le dépôt de modèles **recordschool** est consacré à l'école-chercheurs « Approches interdisciplinaires de la modélisation des agroécosystèmes » organisée par l'**INRA** en mars 2017 à Sainte Foy Lès Lyon. Le simulateur auquel on va s'intéresser est le simulateur **record_school.vpz** du modèle **RS_FilRouge** qui fait partie du dépôt de modèles **recordschool**.

Les services web erecord proposent des requêtes permettant de connaître les simulateurs, les modèles et les dépôts de modèles existants sous erecord. Il s'agit des ressources [db/vpz](#), [db/pkg](#), [db/rep](#)...

La suite de cette annexe donne différents moyens de trouver l'Id d'un simulateur.



interface utilisateur erecord



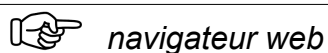
L'interface web utilisateur erecord <http://erecord.toulouse.inra.fr:8000/home> donne en page d'accueil la liste des dépôts de modèles, la liste des modèles, la liste des simulateurs.

L'enchaînement des *requêtes* suivantes guide la sélection du simulateur **record_school.vpz** :

- > sélection du « models repository » **recordschool**
- > sélection du « model » **RS_FilRouge**
- > sélection du « simulator » **record_school.vpz**

Notons que le simulateur **record_school.vpz** a pour **Id=764**.

C'est cet identifiant qui servira à désigner le simulateur dans les requêtes erecord à son sujet.



Il est également possible d'accéder à ces informations depuis un navigateur web sans passer par l'interface web utilisateur erecord, en tapant directement l'URL de la requête dans la barre d'adresse du navigateur.

Requête

> saisie de l'adresse <http://erecord.toulouse.inra.fr:8000/db/rep/>

Réponse

On obtient la liste des « models repositories ».

Notons que le dépôt de modèles recordschool a pour Id=8.

Requête

> saisie de l'adresse <http://erecord.toulouse.inra.fr:8000/db/pkg/?rep=8>

Réponse

On obtient la liste des « models » du « models repository » qui a pour Id=8 (recordschool).

Notons que le modèle RS_FilRouge a pour Id=184.

Requête

> saisie de l'adresse <http://erecord.toulouse.inra.fr:8000/db/pkg/184/>

Réponse

On obtient la liste des « simulators » du « model » qui a pour Id=184 (RS_FilRouge).

Notons que le simulateur record_school.vpz a pour Id=764.

La *requête* suivante permet de vérifier l'identifiant d'un simulateur :

> saisie de l'adresse <http://erecord.toulouse.inra.fr:8000/db/vpz/764/name>

Réponse


On obtient le nom du simulateur qui a pour Id=764.

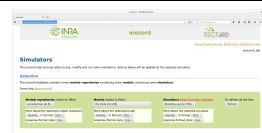
Le simulateur record_school.vpz a bien pour Id=764.

Annexe B. Observer un simulateur

Les services web erecord permettent d'obtenir des informations d'un simulateur, concernant principalement ses conditions de simulation : paramétrage , début et durée de simulation.

La ressource erecord correspondante est [vpz/input](#).

 *interface utilisateur erecord*




Requête sous l'interface web utilisateur erecord <http://erecord.toulouse.inra.fr:8000/home> après avoir sélectionné le simulateur **record_school.vpz (Id=764)** :

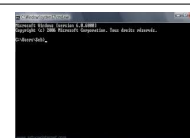
```
> bouton « See » (To see the simulator)
```

Réponse

On obtient les informations du simulateur **record_school.vpz** :

- La page résultat donne les valeurs de début **Begin** et durée **Duration** de simulation (rubrique « General information ») ainsi que le nom et la valeur initiale de chacun des **paramètres** du simulateur (rubrique « Parameters values »).
- La page résultat donne également (rubrique « Selection information ») les **identifiants** des « **conditions** », « **parameters** », « **views** » et « **output datas** » qui serviront quand il s'agira de les sélectionner lors de certaines requêtes.

 *interface en ligne de commande*



Requête depuis une interface en ligne de commande

```
> curl -H "Content-Type: application/x-www-form-urlencoded" -d  
'vpz=764&style=compactlist&outselect=all&format=json'  
http://erecord.toulouse.inra.fr:8000/vpz/input/
```

Remarque : pour une meilleure lisibilité du résultat, demander un résultat au format yaml plutôt que json ; pour cela remplacer dans la requête format=json par **format=yaml**.

Paramètre de requête **parselect** :

On voit parmi tous les paramètres du simulateur **record_school.vpz** retournés dans la réponse à la requête **vpz/input** d'observation du simulateur, que certains de ces paramètres ont une signification agronomique et que d'autres sont des éléments de construction du modèle sans intérêt sur le plan agronomique, comme par exemple `cond_DE.mode`, `cond_DE.time-step`, `cond_DE_LeafIndexArea.variables`, `cond_DE_LeafIndexArea_grassland.variables`...

Si l'on ne souhaite recevoir dans la réponse que des paramètres ayant une signification agronomique, on peut utiliser le paramètre de requête **parselect** qui permet de filtrer certains paramètres retournés : voir exemple dans « B. Observer un simulateur » du [document](#).

Le paramètre de requête **parselect** peut valoir des noms de sélection de paramètres ou des noms de sélection de conditions (groupes de paramètres), que l'on a moyen de connaître par la requête **vpz/input** d'observation du simulateur : voir ci-dessus « Annexe A. Observer un simulateur » où il s'agit des identifiants des « parameters » et « conditions » donnés dans la rubrique « Selection information ».

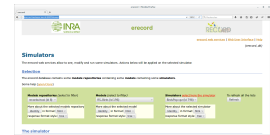
Annexe C. Simuler

Les services web erecord permettent de modifier un simulateur (son paramétrage...) avant d'en lancer la simulation et d'en obtenir les résultats de simulation.

La ressource erecord correspondante est [vpz/output](#).



interface utilisateur erecord



L'interface web utilisateur erecord <http://erecord.toulouse.inra.fr:8000/home> donne les résultats de simulation dans un tableau ou sous forme graphique.

Requête avec laquelle le simulateur est lancé tel que (sans en modifier les conditions) sous l'interface web utilisateur erecord <http://erecord.toulouse.inra.fr:8000/home>, après avoir sélectionné le simulateur de son choix :

```
> bouton « Run » (To run the simulator)
```

Réponse

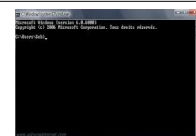
On obtient les résultats de simulation du simulateur choisi :

- Résultats numériques (bouton Numeric / « **Show** »).
- Résultats graphiques (bouton « **Add Y(X)** » après avoir choisi « **X** » et « **Y** »).

Remarque : la demande de simulation au moyen de l'interface web utilisateur erecord <http://erecord.toulouse.inra.fr:8000/home> (bouton « Run ») ne fonctionne pas pour le simulateur **record_school.vpz (Id=764)**, probablement à cause de sa quantité totale de données de sortie. Cette simulation peut être lancée d'autres manières : voir ci-dessous et voir « C. Simuler » du [document](#).



interface en ligne de commande



Il est possible de lancer la simulation depuis une interface en ligne de commande.

Requête avec laquelle le simulateur est lancé tel que (sans en modifier les conditions) :

```
> curl -H "Content-Type: application/x-www-form-urlencoded" -d  
'vpz=764&style=compact&plan=single&restype=dataframe&format=yaml&outselect=all'  
http://erecord.toulouse.inra.fr:8000/vpz/output/
```

Dans la requête suivante :

- le simulateur est modifié avant simulation. Pour cela on utilise d'une part le paramètre de requête **duration** qui permet de modifier la durée de simulation et d'autre part des paramètres de requête qui permettent de modifier le paramétrage. Ces paramètres de requête sont les noms de sélection des paramètres que l'on a moyen de connaître par la requête **vpz/input** d'observation du simulateur : voir « Annexe A. Observer un simulateur » où il s'agit des identifiants des « parameters » donnés dans la rubrique « Selection information ».

- pour ne garder que les données de sortie qui nous intéressent, on utilise le paramètre de requête **outselect** qui permet de filtrer les données de sortie retournées. Le paramètre de requête outselect peut valoir des noms de sélection de données de sortie ou des noms de sélection de vues (groupes de données de sorties), que l'on a moyen de connaître par la requête **vpz/input** d'observation du simulateur : voir « Annexe A. Observer un simulateur » où il s'agit des identifiants des « output datas » et « views » donnés dans la rubrique « Selection information ».

Requête

```
> curl -H "Content-Type: application/x-www-form-urlencoded" -d
'vpz=764&duration=10&cond_DE_River.WatershedArea=800.0&cond_DE_Dam.MaximumVolume=10000.0&style=compact&plan=single&restype=dataframe&format=json&outselect=Dam.Top
model:Dam.Volume&outselect=Dam.Top model:Dam.RainToDam&outselect=Meteo.Top
model,Climate:Climate.Rain' http://erecord.toulouse.inra.fr:8000/vpz/output/
```

Réponse

On obtient les résultats de simulation du simulateur **record_school.vpz** :

```
{"res":{"\"Dam.Top model:Dam.Volume\": [900.000011920929, 883.000011920929,
444.4742334356488, 348.33085642060473, 313.33085642060473, 292.33085642060473,
272.33085642060473, 243.33085642060473, 200.33085642060473, 165.33085642060473,
162.33085642060473], \"Dam.time\": [2454024.0, 2454025.0, 2454026.0, 2454027.0, 2454028.0,
2454029.0, 2454030.0, 2454031.0, 2454032.0, 2454033.0, 2454034.0], \"Meteo.Top
model,Climate:Climate.Rain\": [0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0, 1.4], \"Meteo.time\":
[2454024.0, 2454025.0, 2454026.0, 2454027.0, 2454028.0, 2454029.0, 2454030.0, 2454031.0,
2454032.0, 2454033.0, 2454034.0], \"Dam.Top model:Dam.RainToDam\": [0.0, 0.0, 0.0, 0.0, 0.0,
0.0, 0.0, 0.0, 0.0, 0.0, 28.0]}},\"plan\":\"single\",\"restype\":\"dataframe\"}
```

Remarque : pour une meilleure lisibilité du résultat, demander un résultat au format yml plutôt que json ; pour cela remplacer dans la requête format=json par **format=yaml**.

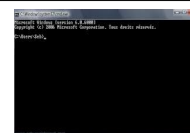
Annexe D. Travailler par fichier xls

Les services web erecord permettent de travailler par fichier xls pour modifier et simuler le simulateur.

La ressource erecord correspondante est [vpz/experiment](#) ([GET](#), [POST](#)).



interface en ligne de commande



Il est possible de travailler par fichier xls depuis une interface en ligne de commande.

Le paramètre de requête `mode=todownload` permet de rendre le résultat téléchargeable.

Requête permettant de recevoir le fichier xls contenant l'état initial du simulateur (paramètres...) :

```
> curl "http://erecord.toulouse.inra.fr:8000/vpz/experiment/?vpz=764&mode=todownload"
```

Réponse

```
"The experiment report can be downloaded at :  
http://erecord.toulouse.inra.fr:8000/media/slm/erecord\_vpz/download/20170307\_132714\_4b0c9029-6073-413b-a1b1-f1e2772b79fd/\_24222\_input\_folder.zip"
```

=> Le fichier xls (contenu dans le .zip) téléchargé, contenant l'état initial du simulateur, est sauvegardé localement : [experiment_RS_FilRouge_initial.xls](#) .

Opérations locales : après en avoir modifié les conditions de simulation (paramètres, durée...), le fichier xls modifié est sauvegardé localement : [experiment_RS_FilRouge_modifie.xls](#) .

Requête permettant de lancer la simulation basée sur le fichier xls modifié et de recevoir le fichier xls contenant à la fois les résultats de simulation et les conditions de simulation :

```
> curl -F "experimentfile=@experiment\_RS\_FilRouge\_modifie.xls" -F "vpz=764" -F  
"mode=todownload" http://erecord.toulouse.inra.fr:8000/vpz/experiment/
```

Réponse

```
"The experiment report can be downloaded at :  
http://erecord.toulouse.inra.fr:8000/media/slm/erecord\_vpz/download/20170307\_133030\_fe53544c-9d65-4c6d-b6de-a1d10a402e2d/\_24223\_input\_folder.zip"
```

=> Le fichier xls (contenu dans le .zip) téléchargé, contenant les conditions de simulation et les résultats de simulation, est sauvegardé localement : [experiment_RS_FilRouge_simu.xls](#) .