

# ORCHIDS OSC PROTOCOL

## TABLE OF CONTENTS

<b>Introduction.....</b>	<b>3</b>
<b>Session.....</b>	<b>3</b>
Initialisation .....	3
Fin de la session .....	3
Set Criteria .....	4
Orchestrate .....	4
<b>Database.....</b>	<b>5</b>
Set Cache Size.....	5
Set Cache Size Default .....	5
Add Sound Directories.....	6
Remove Sound Directory .....	6
Get Symbolics.....	6
Get Sounds Query .....	6
Get Descriptor Min Max .....	7
Get Single .....	7
Get Index List .....	8
<b>Production .....</b>	<b>8</b>
Set Resolution .....	8
Set Orchestra .....	8
Set Filters .....	9
Reset Filters .....	9
set Harmonic Filtering .....	9
Get Score Order.....	10
<b>Target .....</b>	<b>10</b>
Set Soundfile .....	10
Set Analysis Parameters .....	10
Set Abstract Target.....	11
Set Target Duration .....	11
Set Target Feature .....	11
<b>Search.....</b>	<b>12</b>
Set Search Parameter .....	12
<b>Export.....</b>	<b>12</b>

Export Solutions .....	12
Annexe .....	13
Criteria .....	13
Families & instruments.....	14
Search parameters .....	14

# INTRODUCTION

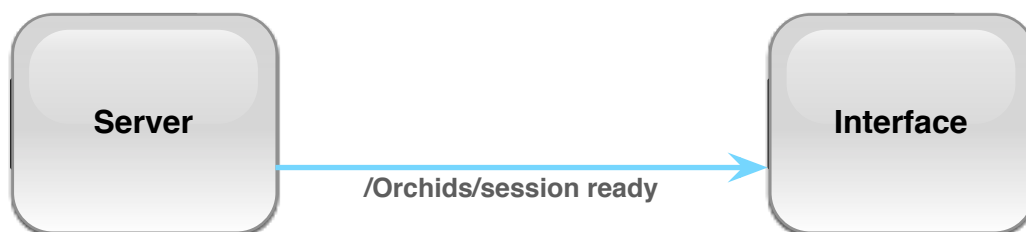
Chaque partie correspond à un des nouveaux modules qui a été implémenté. Tous les message OSC lui correspondant sont expliqués dans les sous-parties. La gestion des erreurs a été modifiée, à chaque fois qu'une erreur est lancée, le serveur va envoyer le message suivant :

`/Orchids/reply/Error` « le message d'erreur »

## SESSION

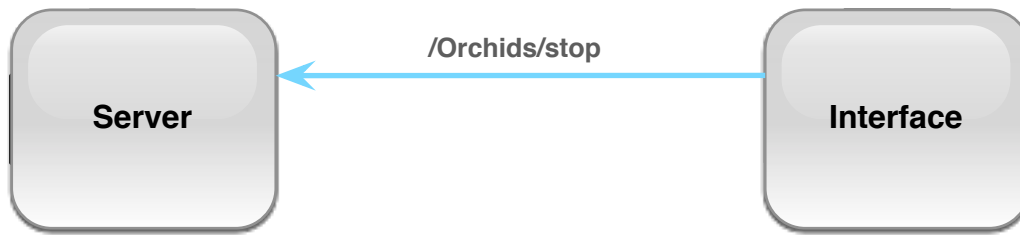
### INITIALISATION

Afin de démarrer Orchids, il faut lancer l'exécutable « Search ». L'application va se connecter automatiquement à la database puis charger les données qu'elle contient, ce module s'appelle « Knowledge ». Une fois cette opération terminée, l'application envoie un message OSC pour informer l'interface que le serveur est prêt.



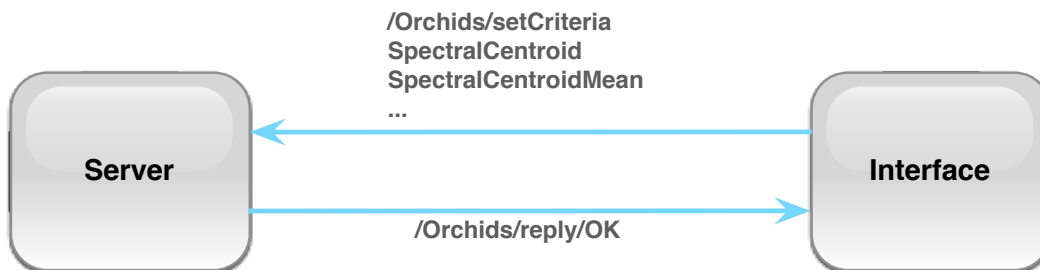
### FIN DE LA SESSION

L'arrêt du serveur s'effectue de la même manière que pour le module Database. L'arrêt d'une session peut prendre quelques secondes le temps que la database se ferme correctement, donc comme pour le module d'avant, il faut garder la bouton « Fermer » ou alors lancer le serveur en parallèle afin que sa fermeture se fasse indépendamment de celle de l'interface.



### SET CRITERIA

Ce message permet d'enregistrer une liste de critères pour une orchestration. Les critères disponibles sont dans l'annexe 1. Si un critère n'est pas conforme, le message d'erreur suivant est envoyé: **"Session:setCriteriaList, Unrecognized criteria : ... "**



### ORCHESTRATE

Ce message lance une orchestration à partir des paramètres donnés. Les conditions pour que l'orchestration se lance sont :

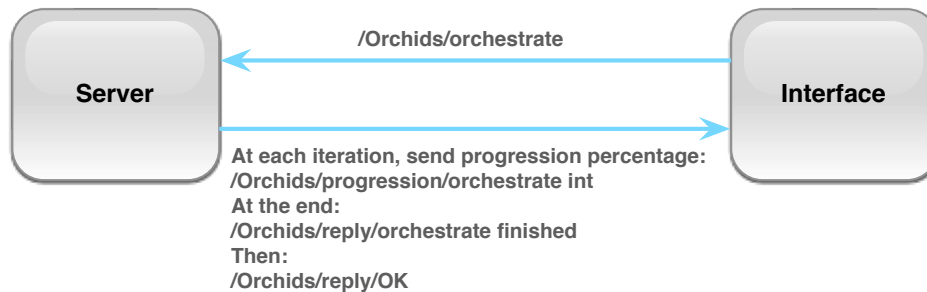
- La liste des critères avec au moins un critère
- La liste des instruments avec au moins deux instruments
- Une cible (fichier ou abstraite)

Les erreurs qui peuvent être lancées :

- **"SearchGenetic::LaunchSearch, Criteria list is empty!"**
- **"SearchGenetic::LaunchSearch, Instruments list is empty!"**
- **"Production:emptySearchSpace, Search space is empty ! Check filters and orchestra"**
- **"Population:generateRandomPopulation:ImpossibleOperation, Search space is empty !"**

L'erreur Production:emptySearchSpace signifie que les filtres appliqués à la database (voir partie Production) n'ont donnés aucun résultat. La dernière erreur signifie qu'il y a eu des résultats, mais un ou plusieurs instruments n'ont aucun son qui correspond aux filtres de recherche.

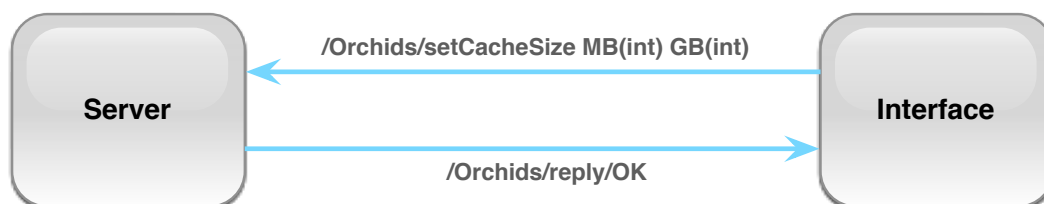
Le message qui indique la progression est envoyé à chaque itération de la recherche et informe l'interface de son avancement en pourcentage. Je pensais que ça pourrait être intéressant de monter une barre de chargement avec le pourcentage dans l'interface.



## DATABASE

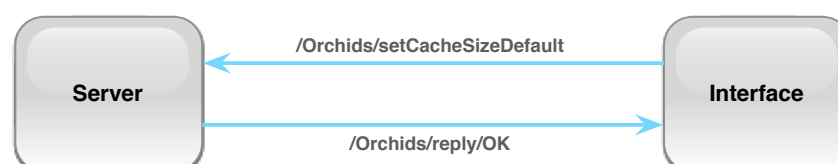
### SET CACHE SIZE

Il y a maintenant la possibilité de choisir la taille du cache de la database. La nouvelle version actuelle de la database fait 4Go, c'est donc le cache maximum. Il y a deux informations à fournir afin de changer la taille de cache, la taille en MB et celle en GB, exemple : 512 et 2. Il y aura alors un cache de 2GB et 512MB. La taille en MB doit être idéalement une puissance de 2, si ce n'est pas le cas je fais de toute façon la conversion après. Ces informations sont ensuite sauvegardées dans un fichier qui est lu au lancement de la base de données, donc le cache restera le même à chaque lancement de la database. Par contre, pour que cette modification prenne effet, je dois fermer et rouvrir la database. Mais Berkeley DB dit « une fois l'environnement fermé, on ne peut pas le ré-ouvrir » ... Il faut donc fermer l'application (et le serveur avec) puis tout relancer pour que la modification prenne effet.



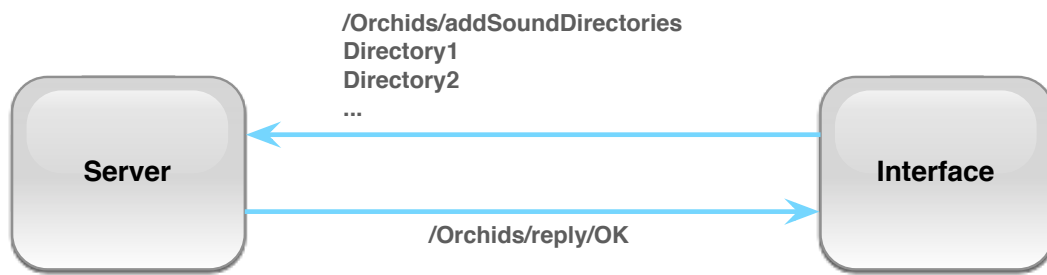
### SET CACHE SIZE DEFAULT

Si l'utilisateur veut reset la taille du cache, le serveur va supprimer le fichier qui sauvegarde les anciennes valeurs. Par la suite, la taille du cache par défaut est 512MB. Comme pour le message d'avant, afin que la modification prenne effet, il faut redémarrer l'appli.

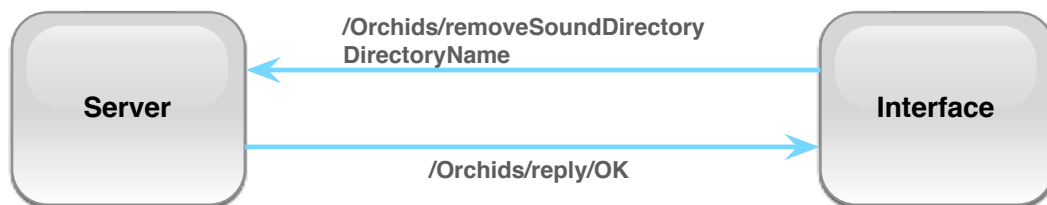


### ADD SOUND DIRECTORIES

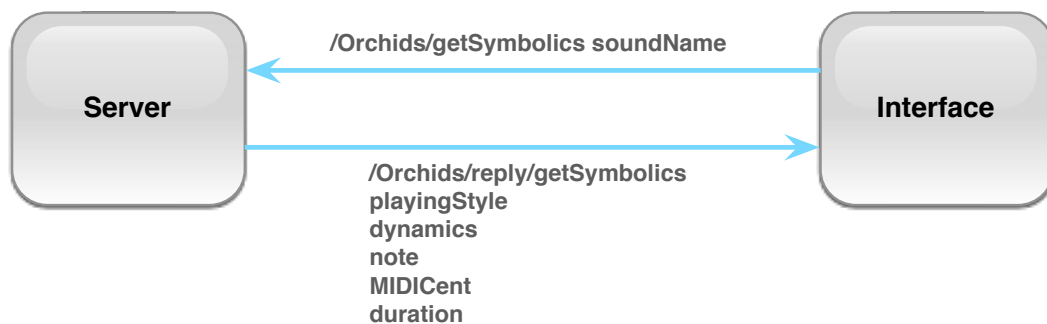
Les messages suivants sont les mêmes que dans le module Database.



### REMOVE SOUND DIRECTORY

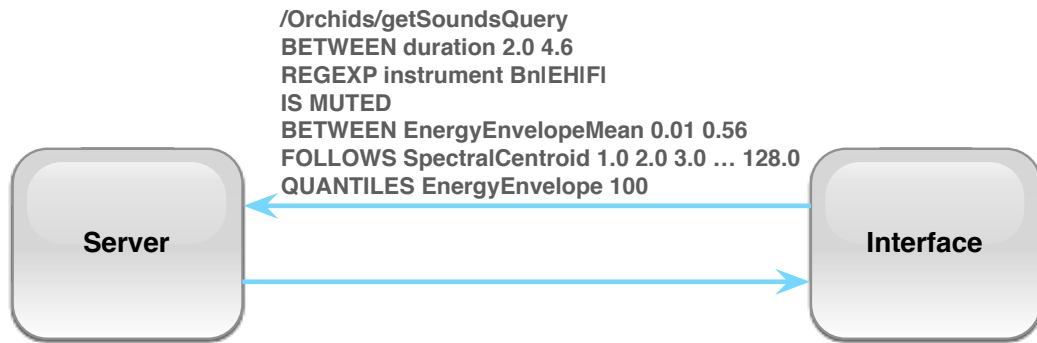


### GET SYMBOLICS



### GET SOUNDS QUERY

Je n'ai pas encore fait ce qu'on s'était dit sur les quantiles pré-calculés, j'attends d'avoir une version plus définitive de SQL avant.



If there are results:  
`/Orchids/reply/getSoundsQuery soundName1 soundName2 ... soundNameN`  
`/Orchids/reply/getSoundsQuery soundName15 soundName16 ... soundNameN`  
`/Orchids/reply/getSoundsQuery END`

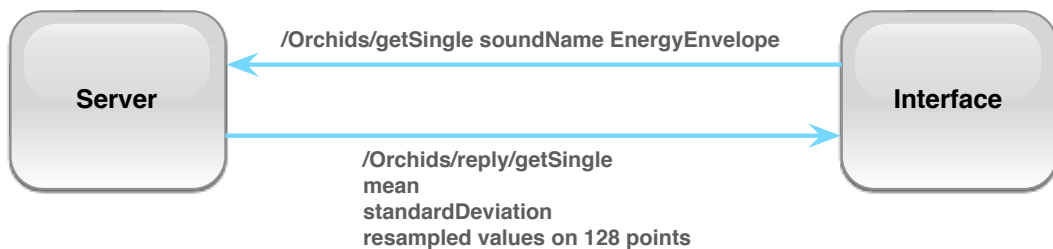
If no results:  
`/Orchids/reply/getSoundsQuery No Data`  
`/Orchids/reply/getSoundsQuery END`

If QUANTILES, `/Orchids/reply/getSoundsQuery` (as above) then:  
`/Orchids/reply/getDescriptorFilteredMinMaxQuantiles`  
`min_Mean max_Mean quantile1 quantile2 ... quantileN`

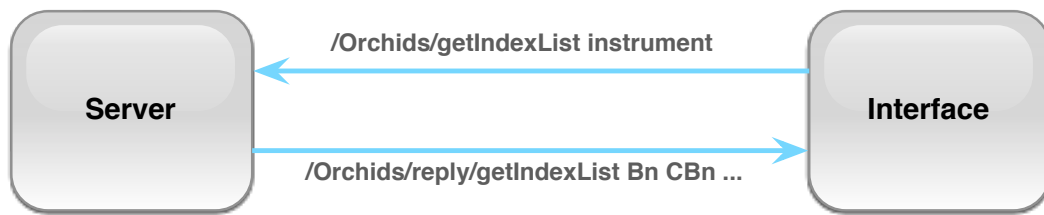
## GET DESCRIPTOR MIN MAX



## GET SINGLE



## GET INDEX LIST

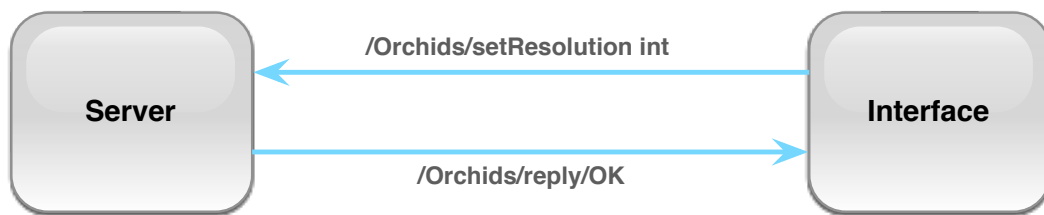


## PRODUCTION

### SET RESOLUTION

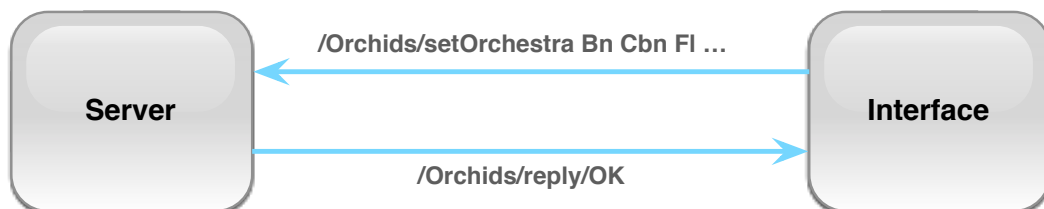
Ce message permet de régler la résolution microtonique. Les valeurs acceptées sont 1, 2, 4 et 8. L'appli fait de toute manière un check et envoie l'erreur suivante si ce n'est pas bon :

**"Production:setResolution:BadResolution, Allowed values for microtonic resolution: 1 2 4 8 (fractions of semitones)"**



### SET ORCHESTRA

Fourni la liste des instruments qui seront utilisés pour l'orchestration. Les familles et les instruments disponibles sont présentés en Annexe 2.

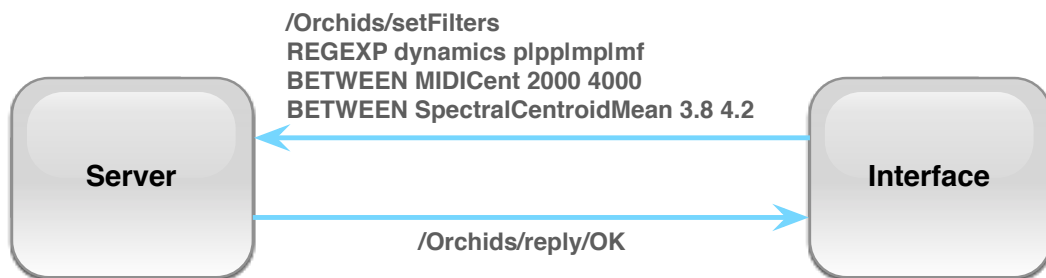




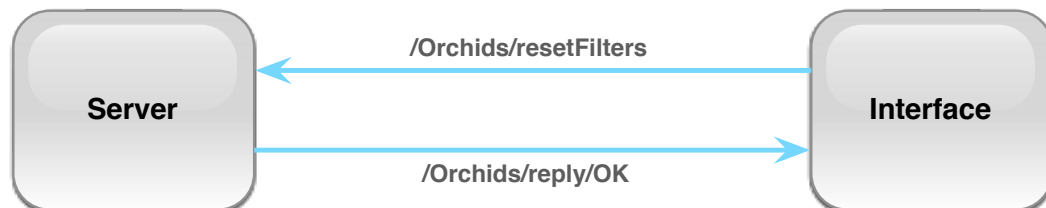
## SET FILTERS

J'ai différencié les requêtes sur la database et les filtres. Les requêtes sur la database permettent plus de libertés, notamment de choisir des instruments qui ne seront pas dans l'orchestration, d'utiliser la requête « FOLLOWS », ...

Les filtres par contre s'applique directement à la production. Le choix des instruments se fait par le biais du message setOrchestra. Les filtres s'écrivent de la même manière que les requêtes mais certains requêtes ne sont plus possible au niveau des filtres comme « FOLLOWS » ou « IS MUTED/UNMUTED » (pour celui là je vais l'ajouter par la suite). En gros, mon idée pour l'interface est de faire un onglet Database qui marche comme pour le module précédent mais de rajouter en dessous un bouton « Apply filters on Orchestration », avec un deuxième bouton « Reset Filters » (voir sous-partie suivante). Dans les filtres seraient envoyés juste les requêtes sur les dynamics, MIDICent, duration, playingStyle, source et tous ce qui concerne les moyennes des descripteurs.

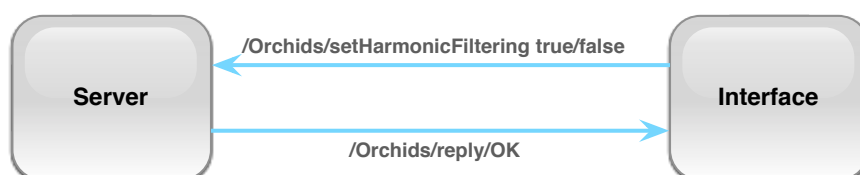


## RESET FILTERS



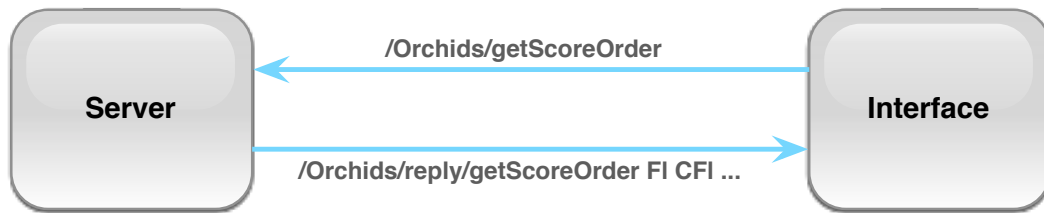
## SET HARMONIC FILTERING

Par défaut, le filtrage harmonique est activé. Ce message permettrait de le désactiver mais je ne suis pas encore sûr de l'utilité qui ça aurait sur l'application. Le filtrage harmonique permet de réduire l'espace de recherche en ne gardant que les sons donc les fréquences des partiels correspondent à celles de la cible.



## GET SCORE ORDER

Permet de récupérer la liste des instruments présents dans la database de manière « ordonnée », dans le style d'un orchestre. Je pense que ça sera utile lors de l'affichage des solutions.

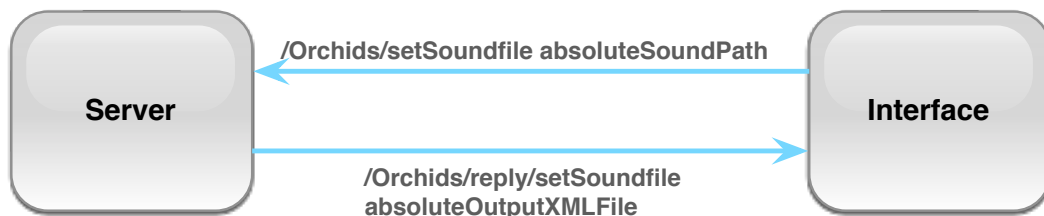


## TARGET

### SET SOUNDFILE

Fournir le chemin absolu de la cible puis l'application va l'analyser. Les résultats de l'analyse sont sous la forme d'un fichier XML donc le chemin est envoyé par message OSC `(/tmp/soundfile.wav.xml)`. Pour le parse du XML, voir le module d'Analyse de Philippe.

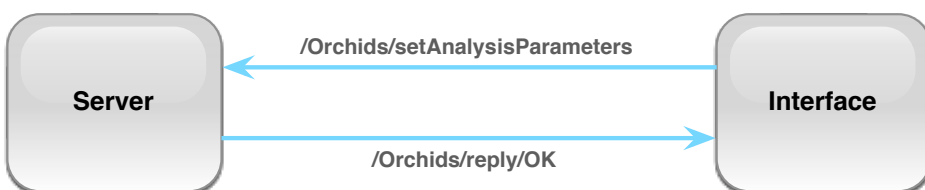
Message d'erreur possible (mais ça ne risque pas d'arriver...) : **"Error while opening XML output directory"**



### SET ANALYSIS PARAMETERS

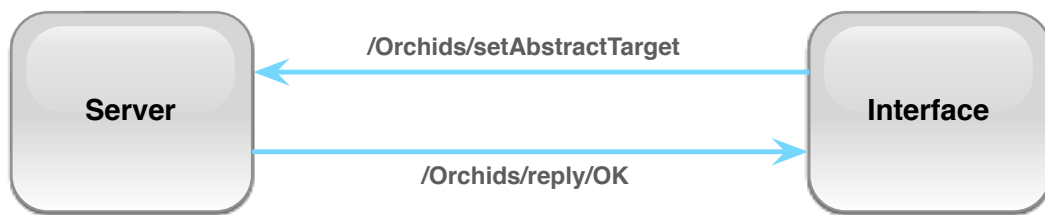
A envoyé avant `setSoundfile`, par la suite, l'analyse de la cible prendra en compte le fichier de configuration « `Analysis.cfg` ». Je ne suis pas sûr, mais Philippe avait dû faire une petite interface pour éditer ce fichier de configuration.

Message d'erreur possible : **"Value error in configuration file. Parameters ... is out of range"**



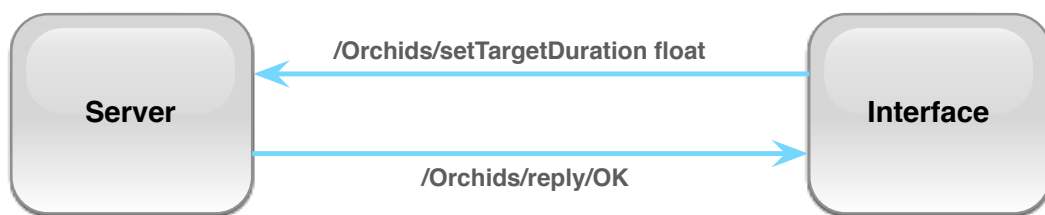
### SET ABSTRACT TARGET

La session en cours va utiliser une cible abstraite.



### SET TARGET DURATION

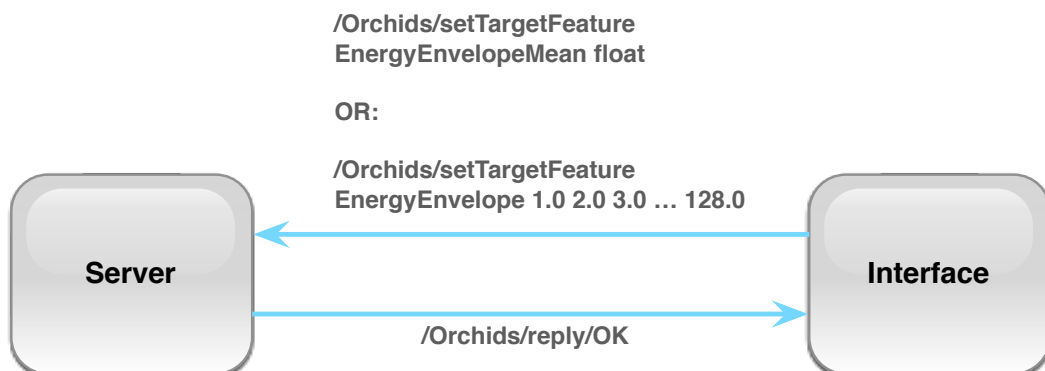
Utile dans le cas d'une cible abstraite, permet de régler sa durée.



### SET TARGET FEATURE

Dans le cas d'une cible abstraite, il faut que certains descripteurs aient des valeurs sinon l'orchestration n'a aucun sens, notamment ceux présents dans la liste des critères. Il y a donc la possibilité de changer la valeur d'un descripteur (sur 128 points) ou de changer la moyenne d'un descripteur.

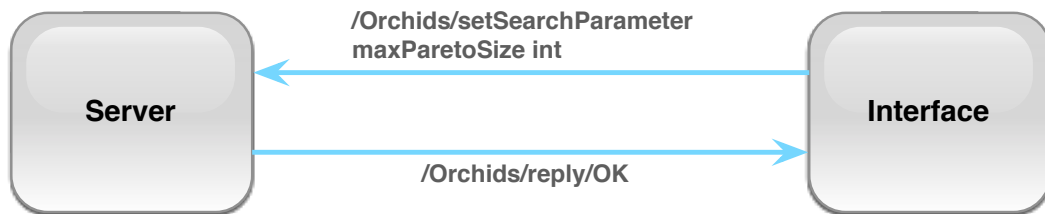
Message d'erreur possible : **"TargetSound:getFeature:Unknown, ... is not a target feature"**



## SEARCH

### SET SEARCH PARAMETER

Permet de modifier un paramètre de la recherche. Les paramètres disponibles sont en Annexe 3.



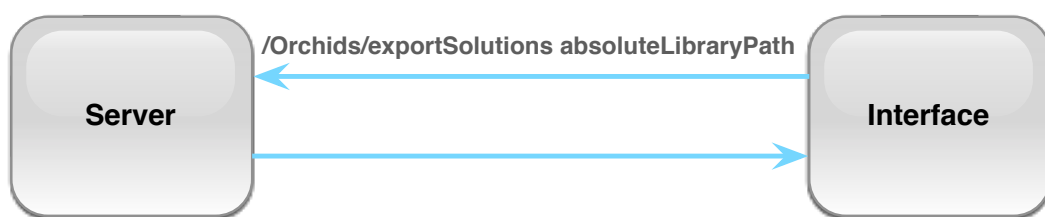
## EXPORT

### EXPORT SOLUTIONS

Pour le moment, le seul export que j'ai implémenté est un export de chaque solution en fichier .wav. J'implémenterai par la suite d'autres exports.

Pendant que l'exportation est en cours, à chaque fois qu'un son a été exporté, j'envoie un message de la forme « Export solutions 2/13 ». L'idée encore une fois c'est de pouvoir l'afficher sur l'interface. Ensuite, le serveur envoie la liste des fichiers wav de sortie qui seront dans le dossier `/tmp/solutionTmp/`.

En paramètre du message, il faut fournir le chemin absolu du dossier SOL (p.e. `/Applications/Orchids/SOL/`).



For each sound:  
`/Orchids/progression/exportSolutions` `string(n/nbSolutions)`

Then:  
`/Orchids/reply/exportSolutions` `absoluteOutputWAVFile1`  
`absoluteOutputWAVFile2 ...`

# ANNEXE

## CRITERIA

"EnergyEnvelope",  
"EnergyEnvelopeMean",  
"HarmonicEnergy",  
"HarmonicEnergyMean",  
"HarmonicOddToEvenRatio",  
"HarmonicOddToEvenRatioMean",  
"HarmonicSpectralCentroid",  
"HarmonicSpectralCentroidMean",  
"HarmonicSpectralDecrease",  
"HarmonicSpectralDecreaseMean",  
"HarmonicSpectralDeviation",  
"HarmonicSpectralDeviationMean",  
"HarmonicSpectralKurtosis",  
"HarmonicSpectralKurtosisMean",  
"HarmonicSpectralRolloff",  
"HarmonicSpectralRolloffMean",  
"HarmonicSpectralSkewness",  
"HarmonicSpectralSkewnessMean",  
"HarmonicSpectralSlope",  
"HarmonicSpectralSlopeMean",  
"HarmonicSpectralSpread",  
"HarmonicSpectralSpreadMean",  
"HarmonicSpectralVariation",  
"HarmonicSpectralVariationMean",  
"Inharmonicity",  
"InharmonicityMean",  
"Loudness",  
"LoudnessMean",  
"NoiseEnergy",  
"NoiseEnergyMean",  
"Noisiness",  
"NoisinessMean",  
"PerceptualOddToEvenRatio",  
"PerceptualOddToEvenRatioMean",  
"PerceptualSpectralCentroid",  
"PerceptualSpectralCentroidMean",  
"PerceptualSpectralDecrease",  
"PerceptualSpectralDecreaseMean",  
"PerceptualSpectralDeviation",  
"PerceptualSpectralDeviationMean",  
"PerceptualSpectralKurtosis",  
"PerceptualSpectralKurtosisMean",  
"PerceptualSpectralRolloff",  
"PerceptualSpectralRolloffMean",  
"PerceptualSpectralSkewness",  
"PerceptualSpectralSkewnessMean",  
"PerceptualSpectralSlope",  
"PerceptualSpectralSlopeMean",  
"PerceptualSpectralSpread",  
"PerceptualSpectralSpreadMean",  
"PerceptualSpectralVariation",  
"PerceptualSpectralVariationMean",  
"Sharpness",  
"SharpnessMean",  
"SignalZeroCrossingRate",  
"SignalZeroCrossingRateMean",  
"SpectralCentroid",  
"SpectralCentroidMean",  
"SpectralDecrease",  
"SpectralDecreaseMean",  
"SpectralKurtosis",  
"SpectralKurtosisMean",  
"SpectralRolloff",  
"SpectralRolloffMean",  
"SpectralSkewness",  
"SpectralSkewnessMean",  
"SpectralSlope",

```

"SpectralSlopeMean",
"SpectralSpread",
"SpectralSpreadMean",
"SpectralVariation",
"SpectralVariationMean",
"Spread",
"SpreadMean",
"TotalEnergy",
"TotalEnergyMean",
"PartialsAmplitude",
"PartialsMeanAmplitude",
"PartialsEnergy",
"partialsMeanEnergy",
"loudnessFactor"

```

## FAMILIES & INSTRUMENTS

"Flutes",	"Piccolo",	"Picc"
"Flutes",	"Flute",	"Fl"
"Flutes",	"Bass-Flute",	"BFl"
"Flutes",	"Contrabass-Flute",	"CbFl"
"Oboes",	"Oboe",	"Ob"
"Oboes",	"English-Horn",	"EH"
"Clarinets",	"Clarinet-Eb",	"ClEb"
"Clarinets",	"Clarinet-Bb",	"ClBb"
"Clarinets",	"Bass-Clarinet-Bb",	"BClBb"
"Clarinets",	"Contrabass-Clarinet-Bb",	"CbClBb"
"Bassoons",	"Bassoon",	"Bn"
"Bassoons",	"Contrabassoon",	"Cbn"
"Saxophones",	"Alto-Sax",	"ASax"
"Horns",	"Horn",	"Hn"
"Trumpets",	"Trumpet-C",	"TpC"
"Trombones",	"Tenor-Trombone",	"TTbn"
"Trombones",	"Bass-Trombone",	"BTbn"
"Tubas",	"Bass-Tuba",	"BTb"
"Tubas",	"Contrabass-Tuba",	"CbTb"
"Keyboards",	"Accordion",	"Acc"
"Plucked-Strings",	"Harp",	"Hp"
"Strings",	"Violin",	"Vn"
"Strings",	"ViolinEnsemble",	"Vns"
"Strings",	"Viola",	"Va"
"Strings",	"ViolaEnsemble",	"Vas"
"Strings",	"Violoncello",	"Vc"
"Strings",	"VioloncelloEnsemble",	"Vcs"
"Strings",	"Contrabass",	"Cb"
"Strings",	"ContrabassEnsemble",	"Cbs"

## SEARCH PARAMETERS

```

int initPopSize      = 200; // Initial population size
int maxPopSize       = 400; // Maximal population size
int matingPoolSize   = 50;  // Size of mating pool

```

```
int paretoMaxSize = 50; // Maximum size of pareto front  
int nIter          = 100; // Number of evolution iterations
```