

# ***Spécification des interactions client-serveur du jeu d'échec en version blitz avec brouillard de guerre***

Statut : document de travail

Révision du document : fork blitz – janvier 2016

## **Introduction**

Ce document comporte les spécifications du dialogue client-serveur du jeu d'échec avec brouillard de guerre, en version blitz définissant les noms des web services, leurs paramètres d'entrées et les objets JSON qu'ils renvoient.

Les exemples de données n'ont pas été systématiquement validés.

## **Cycle de vie d'une session d'utilisation du client**

Cycle de vie d'une consultation de partie sur le client :

- L'utilisateur se logue (auth.json.php) et reçoit la liste de ses parties en cours
- L'utilisateur peut créer une nouvelle partie (nouvellePartie.json.php)
- L'utilisateur choisit de consulter une partie
- L'utilisateur demande un CR complet (cr.json.php), le reçoit, et le client affiche la situation au dernier coup joué.
- l'utilisateur joue un coup – il reçoit un rapport de mise à jour de la situation (maj.json.php)
- Le client se met à jour automatiquement ou manuellement (maj.json.php). Si l'autre joueur a joué son coup, le client met à jour son affichage.

Le serveur offre cinq web services :

- cr.json.php => l'historique d'une partie jusqu'au tour courant.
- maj.json.php => la mise à jour de la situation actuelle d'une partie
- parties.json.php => la liste des parties
- register.json.php => s'inscrire sur le serveur
- auth.json.php => authentification qui ouvre une session et renvoie la liste des parties.
- nouvellePartie.json.php => création d'une nouvelle partie ou inscription sur une partie en attente de son deuxième joueur.

## **Format d'échange du WS maj.json.php**

### ***En entrée***

Paramètres (transmis en mode GET) :

Obligatoires :

- "partie" : identifiant de la partie en cours

- "cote" : 1 si c'est le blanc qui fait la requête, 2 si c'est le noir (0 si c'est l'arbitre, valeur par défaut).
- "tour" : numéro du tour. Le premier tour est le tour 1
- "trait" : 1 si c'est aux blancs de jouer, 2 si c'est aux noirs

Optionnel :

- "coup" : rang du coup joué dans le tableau de coups du CR précédent

### **Traitement serveur**

- Le service vérifie que l'identifiant de login (tel qu'il figure dans la session) est valide et correspond au joueur inscrit sur la partie du côté indiqué (paramètres "partie" et "cote").
- Le service vérifie que la partie en est au stade indiqué par les paramètres "tour" et "trait".
- si le paramètre optionnel est fourni et est correct (ie si le tour indiqué est bien le tour courant, si le joueur a effectivement le trait, et si le rang du coup est valide) alors l'arbitrage est lancé pour jouer le coup et rédiger une nouvelle situation
- Si l'une des conditions précédentes n'est pas vérifiée, une erreur sera lancée

### **En sortie**

- si une erreur est détectée, le serveur renvoie : {"erreur": "ici, message d'erreur"}
- si aucune mise à jour n'est disponible, le serveur renvoie : {"ras": 1}
- sinon, le serveur envoie le contenu de la mise à jour.

Après un coup joué par le joueur, la mise à jour est constituée de :

- "je\_joue" : le détail du coup, sous la forme [i0,j0,i1,j1,"option"]. Le dernier paramètre, optionnel, est un paramètre qui indique si un coup est un roque, une prise en passant, ou une promotion (respectivement : "roque", "pp", "X", avec X = D, T, C ou F, code standard de la pièce choisie pour la promotion)
- "vues" : un tableau des cases vues grâce à ce nouveau coup, chaque case étant codée sous la forme : [i,j,"piece"]. Le dernier paramètre, optionnel, est la nature de la pièce éventuellement présente sur la case.
- "propose\_nul" : vaut 1 si le joueur joue en proposant la partie nulle.
- "accepte\_nul" : vaut 1 si après une proposition de nulle, le deuxième joueur l'accepte.
- "abandon" : vaut 1 si le joueur joue en abandonnant la partie.

Sur le client, la liste des cases vues est déterminée à ce stade à partir de la liste des cases vues au tour précédent, plus les cases vues dans le présent CR.

Après un coup joué par le joueur adverse, la mise à jour est constituée de :

- "nature" : la nature de la pièce jouée (R, D, T, F, C ou p). Paramètre absent si les deux cases de départ et d'arrivée du coup sont toutes deux dans le brouillard de guerre.
- "il\_joue" : le détail du coup, selon le même format que précédemment. Si la case de départ (resp. d'arrivée) est dans le brouillard de guerre, les paramètres i0 et j0 (resp. i1 et j1) sont nuls. "option" n'est pas renseignée si la case d'arrivée est dans le brouillard de guerre.

- "coups" : la liste des coups maintenant possibles, sous la forme [x0,y0,x1,y1,"option"]. "option" est un paramètre qui indique si un coup est un roque, une prise en passant, ou une promotion (respectivement : "roque", "pp", "X", avec X = D, T, C ou F, code standard de la pièce choisie pour la promotion)
- "vues" : la liste des cases vues alors qu'aucun coup n'y est possible. Par exemple, un pion qui ne peut pas avancer parce qu'il est bloqué voit la pièce qui le bloque. Ou bien : dans le cadre de la prise en passant, un pion voit le pion qu'il peut prendre (la case vide de sa destination est dans la liste des coups possibles).

Sur le client, la liste des cases vues est déterminée à ce stade à partir de la liste des coups possibles, plus les autres cases vues.

Dans les deux cas, la mise à jour peut contenir l'un des paramètres suivants

- "pat" : vaut 1 si la partie est un pat. Paramètre absent sinon.
- "abandon" : vaut 1 le joueur qui doit jouer abandonne au lieu de jouer. Si ce paramètre est présent, alors il n'y a pas de coup joué.
- XXX proposition d'une partie nulle

## Format d'échange du WS cr.json.php

### **En entrée**

Paramètres (transmis en mode GET) :

Obligatoires :

- "partie" : identifiant de la partie en cours

### **Traitement serveur**

- Le service vérifie que l'identifiant de login (tel qu'il figure dans la session) est valide et correspond à l'un des joueurs inscrits sur la partie (paramètres "partie").
- Il renvoie ensuite soit le message d'erreur, soit le contenu du CR

### **En sortie**

si une erreur est détectée, le serveur renvoie : {"erreur":"ici, message d'erreur"}

Sinon, le serveur renvoie un objet json qui comporte les champs suivants :

- "tour" : numéro du tour. Le premier tour est le tour 1
- "trait" : 1 si c'est aux blancs de jouer, 2 si c'est aux noirs
- "cote" : 1 si c'est le blanc qui fait la requête, 2 si c'est le noir (0 si c'est l'arbitre, valeur par défaut).
- "histo" : tableau des différentes mises-à-jour depuis le début de la partie.

Note : Au tour 1, avant le premier coup des blancs, il n'y a pas encore de mise à jour disponible. Voir l'exemple ci-dessous pour la valeur du premier CR des joueurs blanc et noir.

## Exemples – Session 1

Durant leur première session, blancs et noirs jouent les trois premiers tours.

Tour 1 – les **blancs** ont le trait – CR des **noirs** :

Le client envoie :

cr.json.php?partie=1&cote=2

Le serveur répond :

{"tour":1,"trait":1,"cote":2,"histo":[]}

Tour 1 – les **blancs** ont le trait – CR des **blancs** :

Le client envoie :

cr.json.php?partie=1& &cote=1

Le serveur répond :

{"tour":1,"trait":1,"cote":1,"histo":[{"coups":[[2,1,1,3],[2,1,3,3],[7,1,6,3],[7,1,8,3],[1,2,1,3],[2,2,2,3],[3,2,3,3],[4,2,4,3],[5,2,5,3],[6,2,6,3],[7,2,7,3],[8,2,8,3],[1,2,1,4],[2,2,2,4],[3,2,3,4],[**4,2,4,4**],[5,2,5,4],[6,2,6,4],[7,2,7,4],[8,2,8,4]]]}

Jouons le coup en gras (1 d4)

Le client envoie :

maj.json.php?partie=1&tour=1&trait=1&cote=1&coup=15

Le serveur répond :

{"je\_joue":[**4,2,4,4**], "vues": [[3,5],[4,5],[5,5]]}

Tour 1 – les **noirs** ont le trait – CR des **noirs** :

Le client envoie :

maj.json.php?partie=1&tour=1&trait=2&cote=2

Le serveur répond :

{"il\_joue":[0,0,0,0],"coups":[[2,8,1,6],[2,8,3,6],[7,8,6,6],[7,8,8,6],[1,7,1,6],[2,7,2,6],[3,7,3,6],[4,7,4,6],[5,7,5,6],[6,7,6,6],[7,7,7,6],[8,7,8,6],[1,7,1,5],[2,7,2,5],[3,7,3,5],[**4,7,4,5**],[5,7,5,5],[6,7,6,5],[7,7,7,5],[8,7,8,5]]}

Les noirs jouent le coup en gras (1 ..- d5)

Le client envoie :

maj.json.php?partie=1&tour=1&trait=2&cote=2&coup=16

Le serveur répond :

{"je\_joue":[**4,7,4,5**], "vues": [[3,4],[4,4,"p"],[5,4]]}

Tour 2 – les **blancs** ont le trait – CR des **blancs** :

Le client envoie :

maj.json.php?partie=1&tour=2&trait=1&cote=1

Le serveur répond :

```
{ "nature": "p", "il_joue": [0,0,4,5], "coups": [[2,1,1,3], [2,1,3,3], [2,1,4,2], [7,1,6,3], [7,1,8,3], [1,2,1,3], [2,2,2,3], [3,2,3,3], [5,2,5,3], [6,2,6,3], [7,2,7,3], [8,2,8,3], [1,2,1,4], [2,2,2,4], [3,2,3,4], [5,2,5,4], [6,2,6,4], [7,2,7,4], [8,2,8,4], [3,1,4,2], [3,1,5,3], [3,1,6,4], [3,1,7,5], [3,1,8,6], [4,1,4,2], [4,1,4,3], [5,1,4,2]] }
```

Les blancs jouent le coup en gras (2- Cc3)

Le client envoie :

situation.json.php?partie=1&tour=2&trait=1&cote=1&coup=2

Le serveur répond :

```
{ "je_joue": [2,1,3,3], "vues": [[2,5]] }
```

Tour 2 – les **noirs** ont le trait – CR des **noirs** :

Le client envoie :

maj.json.php?partie=1&tour=2&trait=2&cote=2

Le serveur répond :

```
{ "il_joue": [0,0,0,0], "coups": [[2,8,1,6], [2,8,3,6], [2,8,4,7], [7,8,6,6], [7,8,8,6], [1,7,1,6], [2,7,2,6], [3,7,3,6], [5,7,5,6], [6,7,6,6], [7,7,7,6], [8,7,8,6], [1,7,1,5], [2,7,2,5], [3,7,3,5], [5,7,5,5], [6,7,6,5], [7,7,7,5], [8,7,8,5], [3,8,4,7], [3,8,5,6], [3,8,6,5], [3,8,7,4], [3,8,8,3], [4,8,4,7], [4,8,4,6], [5,8,4,7]], "vues": [[4,4, "p"]] }
```

Les noirs jouent le coup en gras (2 ..- a5)

Le client envoie :

maj.json.php?partie=1&tour=2&trait=2&cote=2&coup=13

Le serveur répond :

```
{ "je_joue": [1,7,1,5], "vues": [[1,4], [2,4]] }
```

Tour 3 – les **blancs** ont le trait – CR des **blancs** :

Le client envoie :

maj.json.php?partie=1&tour=3&trait=1&cote=1

Le serveur répond :

```
{ "il_joue": [0,0,0,0], "coups": [[3,3,2,1], [3,3,1,4], [3,3,2,5], [3,3,4,5, "p"], [3,3,5,4], [7,1,6,3], [7,1,8,3], [1,2,1,3], [2,2,2,3], [5,2,5,3], [6,2,6,3], [7,2,7,3], [8,2,8,3], [1,2,1,4], [2,2,2,4], [5,2,5,4], [6,2,6,4], [7,2,7,4], [8,2,8,4], [3,1,4,2], [3,1,5,3], [3,1,6,4], [3,1,7,5], [3,1,8,6], [4,1,4,2], [4,1,4,3], [5,1,4,2]] }
```

Les blancs jouent le coup en gras (2- Cxd5)

Le client envoie :

situation.json.php?partie=1&tour=2&trait=1&cote=1&coup=3

Le serveur répond :

```
{ "je_joue": [3,3,4,5], "vues": [[2,6], [3,7, "p"], [5,7, "p"], [6,6]] }
```

Tour 3 – les **noirs** ont le trait – CR des **noirs** :

Le client envoie :

maj.json.php?partie=1&tour=3&trait=2&cote=2

Le serveur répond :

```
{ "nature": "C", "il_joue": [0,0,4,5], "coups": [[2,8,1,6],[2,8,3,6],[2,8,4,7],[7,8,6,6],[7,8,8,6],[1,7,1,6],[2,7,2,6],[3,7,3,6],[5,7,5,6],[6,7,6,6],[7,7,7,6],[8,7,8,6],[1,7,1,5],[2,7,2,5],[3,7,3,5],[5,7,5,5],[6,7,6,5],[7,7,7,5],[8,7,8,5],[3,8,4,7],[3,8,5,6],[1,8,1,7],[1,8,1,6],[3,8,6,5],[3,8,7,4],[3,8,8,3],[4,8,4,7],[4,8,4,6],[4,8,4,5,"C"],[5,8,4,7]] }
```

Les noirs jouent le coup en gras (3 ..- Dxd5)

Le client envoie :

```
maj.json.php?partie=1&tour=2&trait=2&cote=2&coup=28
```

Le serveur répond :

```
{ "je_joue": [4,8,4,5], "vues": [[3,4],[2,3],[1,2,"p"],[5,4],[6,3],[7,2,"p"]] }
```

Le joueur blanc s'est déconnecté. Il ne fait pas avancer sa partie. Toutes les tentatives de mise à jour de la part du joueur noir se voit répondre { "ras":1 }. Il se déconnecte à son tour.

Au moment de cette déconnexion, c'est au joueur blanc de jouer.

## Exemples – Session 2

Le lendemain, le joueur blanc se reconnecte. Voici le CR qu'il reçoit

Tour 4 – les **blancs** ont le trait – CR des **blancs** :

Le client envoie :

```
cr.json.php?partie=1& &cote=1
```

Le serveur répond :

```
{ "tour": 4, "trait": 1, "cote": 1, "histo": [{ "coups": [[2,1,1,3],[2,1,3,3],[7,1,6,3],[7,1,8,3],[1,2,1,3],[2,2,2,3],[3,2,3,3],[4,2,4,3],[5,2,5,3],[6,2,6,3],[7,2,7,3],[8,2,8,3],[1,2,1,4],[2,2,2,4],[3,2,3,4],[4,2,4,4],[5,2,5,4],[6,2,6,4],[7,2,7,4],[8,2,8,4]] }, { "je_joue": [4,2,4,4], "vues": [[3,5],[4,5],[5,5]] }, { "nature": "p", "il_joue": [0,0,4,5], "coups": [[2,1,1,3],[2,1,3,3],[2,1,4,2],[7,1,6,3],[7,1,8,3],[1,2,1,3],[2,2,2,3],[3,2,3,3],[5,2,5,3],[6,2,6,3],[7,2,7,3],[8,2,8,3],[1,2,1,4],[2,2,2,4],[3,2,3,4],[5,2,5,4],[6,2,6,4],[7,2,7,4],[8,2,8,4],[3,1,4,2],[3,1,5,3],[3,1,6,4],[3,1,7,5],[3,1,8,6],[4,1,4,2],[4,1,4,3],[5,1,4,2]] }, { "je_joue": [2,1,3,3], "vues": [[2,5]] }, { "il_joue": [0,0,0,0], "coups": [[3,3,2,1],[3,3,1,4],[3,3,2,5],[3,3,4,5,"p"],[3,3,5,4],[7,1,6,3],[7,1,8,3],[1,2,1,3],[2,2,2,3],[5,2,5,3],[6,2,6,3],[7,2,7,3],[8,2,8,3],[1,2,1,4],[2,2,2,4],[5,2,5,4],[6,2,6,4],[7,2,7,4],[8,2,8,4],[3,1,4,2],[3,1,5,3],[3,1,6,4],[3,1,7,5],[3,1,8,6],[4,1,4,2],[4,1,4,3],[5,1,4,2]] }, { "je_joue": [3,3,4,5], "vues": [[2,6],[3,7,"p"],[5,7,"p"],[6,6]] }, { "nature": "D", "il_joue": [0,0,4,5], "coups": [[3,3,2,1],[3,3,1,4],[3,3,2,5],[3,3,4,5,"D"],[3,3,5,4],[7,1,6,3],[7,1,8,3],[1,2,1,3],[2,2,2,3],[3,2,3,3],[5,2,5,3],[6,2,6,3],[7,2,7,3],[8,2,8,3],[1,2,1,4],[3,2,3,4],[5,2,5,4],[6,2,6,4],[7,2,7,4],[8,2,8,4],[3,1,4,2],[3,1,5,3],[3,1,6,4],[3,1,7,5],[3,1,8,6],[4,1,4,2],[4,1,4,3],[5,1,4,2]] } ] }
```

Tour 4 – les **blancs** ont le trait – CR des **noirs** :

Pour sa reconnexion, le joueur noir recevra le CR suivant.

Le client envoie :

```
cr.json.php?partie=1&cote=2
```

Le serveur répond :

```
{ "tour": 4, "trait": 1, "cote": 2, "histo": [{ "il_joue": [0,0,0,0], "coups": [[2,8,1,6],[2,8,3,6],[7,8,6,6],[7,8,8,6],[1,7,1,6],[2,7,2,6],[3,7,3,6],[4,7,4,6],[5,7,5,6],[6,7,6,6],[7,7,7,6],[8,7,8,6],[1,7,1,5],[2,
```

```
7,2,5],[3,7,3,5],[4,7,4,5],[5,7,5,5],[6,7,6,5],[7,7,7,5],[8,7,8,5]]},{ "je_joue":[4,7,4,5],"vues":[[3,4],[4,4,"p"],[5,4]]},{ "il_joue":[0,0,0,0],"coups":[[2,8,1,6],[2,8,3,6],[2,8,4,7],[7,8,6,6],[7,8,8,6],[1,7,1,6],[2,7,2,6],[3,7,3,6],[4,7,4,6],[5,7,5,6],[6,7,6,6],[7,7,7,6],[8,7,8,6],[1,7,1,5],[2,7,2,5],[3,7,3,5],[5,7,5,5],[6,7,6,5],[7,7,7,5],[8,7,8,5],[3,8,4,7],[3,8,5,6],[3,8,6,5],[3,8,7,4],[3,8,8,3],[4,8,4,7],[4,8,4,6],[5,8,4,7]], "vues":[[4,4,"p"]]}, {"je_joue":[1,7,1,5],"vues":[[1,4],[2,4]]},{ "nature":"C","il_joue":[0,0,4,5],"coups":[[2,8,1,6],[2,8,3,6],[2,8,4,7],[7,8,6,6],[7,8,8,6],[1,7,1,6],[2,7,2,6],[3,7,3,6],[5,7,5,6],[6,7,6,6],[7,7,7,6],[8,7,8,6],[1,7,1,5],[2,7,2,5],[3,7,3,5],[5,7,5,5],[6,7,6,5],[7,7,7,5],[8,7,8,5],[3,8,4,7],[3,8,5,6],[1,8,1,7],[1,8,1,6],[3,8,6,5],[3,8,7,4],[3,8,8,3],[4,8,4,7],[4,8,4,6],[4,8,4,5,"C"],[5,8,4,7]]},{ "je_joue":[4,8,4,5],"vues":[[3,4],[2,3],[1,2,"p"],[5,4],[6,3],[7,2,"p"]]]}]}
```

## Format d'échange du WS parties.json.php

### En entrée

Pas de paramètres

### Traitement serveur

Extraction des parties en cours de ce joueur, et des parties auxquelles il peut s'abonner.

### En sortie

Le fichier renvoie un objet json qui comporte les champs suivants :

- "login" : le login du joueur
- "encours" : le tableau des parties en cours
- "nouvelles" : le tableau des nouvelles parties disponibles

Une partie est un objet comportant les champs suivants :

- id : l'identifiant
- nom : le nom de la partie
- tour : le numéro du tour courant
- trait : qui a le trait ? cf. cr.json.php
- cote : suis-je en train de jouer noir ou blanc ? cf. cr.json.php
- j1 : nom du joueur blanc
- j2 : nom du joueur noir

## Format d'échange du WS nouvellePartie.json.php

### En entrée

Paramètres (transmis en mode GET) :

- "nom" : nom d'une nouvelle partie, pour qu'elle soit créée.
- "identifiant" : identifiant de la partie à laquelle je veux m'inscrire

Ces deux paramètres sont exclusifs l'un de l'autre.

### ***Traitement serveur***

Vérification de toutes les cohérences avant d'accepter l'inscription.

### ***En sortie***

Le fichier renvoie un objet json qui comporte les champs suivants :

- "creation" : message de réussite pour la création
- "inscription" : message de réussite pour l'inscription

## **Format d'échange du WS auth.json.php**

### ***En entrée***

Paramètres (transmis en mode POST) :

- "login" : identifiant du joueur
- "pass" : mot de passe

### ***Traitement serveur***

Vérification de la présence du couple login/pass dans la base.

Extraction des parties en cours de ce joueur, et des parties auxquelles il peut s'abonner.

### ***En sortie***

Le serveur renvoie le même résultat que parties.json.php

## **Format d'échange du WS register.json.php**

### ***En entrée***

Paramètres (transmis en mode POST) :

- "login" : identifiant du joueur
- "pass" : mot de passe
- "mail" : adresse mail du joueur

### ***Traitement serveur***

Vérification de l'absence des paramètres login, et mail dans la base.

### ***En sortie***

Le serveur renvoie

- {"erreur": "ici, message d'erreur"} en cas d'erreur (en particulier de doublons)

Sinon le serveur renvoie le même résultat que parties.json.php