Table des matières

Scripting pour le module extension serveur Calendrier

Objets

Event

Tag

Fonctions pour les étiquettes

Calendar.newTag

```
public Tag newTag()
```

Crée une étiquette vide pour une utilisation ultérieure.

JS call:

```
Calendar.newTag()
```

Paramètres :

–

Retourne :

Une étiquette vide

Exceptions :

`SiatelException` - erreur gérée

Calendar.newTag

```
public Tag newTag(DTO tagData)
```

Crée une étiquette vide pour une utilisation ultérieure.

JS call:

```
Calendar.newTag(tagData)
```

Paramètres :

`tagData` - Un DTO (objet de transfert de données) avec les propriétés initiales de l'étiquette

Retourne :

Une étiquette initialisée avec les propriétés fournies

Exceptions :

`SiatelException` - erreur gérée

Calendar.createTag

```
public void createTag(Tag tag)
```

Crée une nouvelle étiquette permanente ; quand la fonction renvoie l'id membre, l'étiquette est initialisée.

JS call:

```
Calendar.createTag(tag)
```

Paramètres :

`tag` - L'étiquette à créer dans le stockage permanent

Exceptions :

`SiatelException` - erreur gérée

Calendar.readTag

```
public Tag readTag(Integer tagId)
```

Retrouve l'étiquette spécifiée dans le stockage permanent.

JS call:

```
Calendar.readTag(tagId)
```

Paramètres :

`tagId` - id de l'étiquette à retrouver dans le stockage permanent

Retourne :

L'étiquette recherchée

Exceptions :

`SiatelException` - erreur gérée

Calendar.updateTag

```
public void updateTag(Tag tag)
```

Met à jour l'étiquette.

JS call:

```
Calendar.updateTag(tag)
```

Paramètres :

`tag` - L'étiquette à mettre à jour dans le stockage permanent

Exceptions :

`SiatelException` - erreur gérée

Calendar.addTag

```
public void addTag(Event event, List tags)
```

Ajoute les étiquettes à l'événement donné

JS call:

```
Calendar.addTag(event, tags)
```

Paramètres :

`event` - L'événement à associer à l'étiquette

`tags` - Une liste d'étiquettes à associer avec l'événement

Exceptions :

`SiatelException` - erreur gérée

Calendar.removeTag

```
public void removeTag(Event event, Tag tag)
```

Enlève une étiquette donnée d'un événement donné.

JS call:

```
Calendar.removeTag(event, tag)
```

Paramètres :

`event` - L'événement concerné par la suppression d'étiquette

`tag` - L'étiquette à enlever pour l'événement donné

Exceptions :

`SiatelException` - erreur gérée

Calendar.deleteTag

```
public void deleteTag(Integer id)
```

Supprime une étiquette du stockage permanent

JS call:

```
Calendar.deleteTag(tagId)
```

Paramètres :

`tagId` - id de l'étiquette à supprimer du stockage permanent

Exceptions :

`SiatelException` - erreur gérée

Calendar.idOfTag

```
public Integer idOfTag(String tagName)
```

Retrouve l'identification d'une étiquette à partir de son nom

JS call:

```
Calendar.idOfTag(tagName)
```

Paramètres :

`tagName` - Le nom de l'étiquette à rechercher

Retourne :

L'id de l'étiquette recherchée

Exceptions :

`SiatelException` - erreur gérée

Calendar.enumerateTags

```
public void enumerateTags(DTO criteria, List tags)
```

Retrouve une liste d'étiquettes qui correspond aux critères donnés

JS call:

```
Calendar.enumerateTags(criteria, list)
```

Paramètres :

`criteria` - les critères de recherche

`list` - la liste permettant d'afficher les étiquettes

Exceptions :

`SiatelException` - erreur gérée

EXEMPLES

CREATEEVENT (EXEMPLE 1)

```
function createEvent() {
    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var today = sdf.format(new java.util.Date());

    var title = "EVENT_" + today;
    var description = "Description_" + today;
    var userIdNumeric = 96;

    var event = Calendar.newEvent();

    event.setTitle(title);
    event.setDescription(description);
    event.setStart(new java.util.Date());
    event.setEnd(new java.util.Date());
    event.setAllDay(1);
    event.setDateUse(new java.util.Date());
    event.setDateCreation(new java.util.Date());
    event.setCreatedBy(userIdNumeric);

    Calendar.createEvent(event);

    return "Id event : " + event.getId();
}
```

CREATEEVENT (EXEMPLE 2)

```
function createEvent2() {
    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var today = sdf.format(new java.util.Date());

    var title = "EVENT_" + today;
```

```
    var description = "Description_" + today;

    var userIdNumeric = 96;
    var start = sdf.format(new java.util.Date());
    var end = start;

    var event = Calendar.newEvent(
          DTOFactory.newDTO(
               "title", title,
               "description", description,
               "userId", userIdNumeric,
               "start",  start,
               "end", end,
               "allDay", "true"));

    Calendar.createEvent(event);

    return "Id event : " + event.getId();
}
```

UPDATEEVENT (EXEMPLE 1)

```
function updateEvent() {
    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var today = sdf.format(new java.util.Date());

    var title = "UPDATE_EVENT_" + today;

    var event = Calendar.newEvent();

    event.setTitle(title);
    event.setStart(new java.util.Date());
    event.setEnd(new java.util.Date());
    event.setAllDay(1);
    event.setDateUse(new java.util.Date());
```

```
    event.setDateCreation(new java.util.Date());

    try {
          Calendar.createEvent(event);

          var description = "Description_" + today;
          event.setDescription(description);

          Calendar.updateEvent(event);
     } catch (e) {
     }

    return "Id event : " + event.getId();
}
```

UPDATEEVENT (EXEMPLE 2)

```
function updateEvent2() {
    var id = 16;

    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var today = sdf.format(new java.util.Date());

    try {
          var event = Calendar.readEvent(id);

          event.setDescription("UPDATE_DESCRIPTION_" + today);

          Calendar.updateEvent(event);
     } catch (e) {
     }

    return "Id event : " + id;
}
```

DELETEEVENT

```
function deleteEvent() {
    var id = 11;

    try {
          Calendar.deleteEvent(id);
     } catch (e) {
     }
}
```

IDOFEVENT

```
function getEventId() {
    var name = "EVENT_02/09/2021 11:24";

    try {
          var id = Calendar.idOfEvent(name);
          return "Id Event : " + id;
     } catch (e) {
     }
}
```

ENUMERATEEVENTS (EXEMPLE 1)

```
function enumerateEvents() {
    var events = new java.util.ArrayList(0);

    var sdfTime = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy");

    var dto = DTOFactory.newDTO(
          "start", "23/08/2021 00:00",
          "end",  "28/08/2021 00:00");
```

```
var json = '({ "eventList" : [ ';

try {
      Calendar.enumerateEvents(dto, events);

      if (events != null && !events.isEmpty()) {
           for (var i = 0; i < events.size(); i++) {
                var event = events.get(i);

                if (i > 0) {
                     json = json + ', ';
                }

                 var allDay = event.getAllDay() == 1 ? true : false;
                var startEvent = allDay
                     ? sdfTime.format(event.getStart())
                     : sdf.format(event.getStart());
                var endEvent = allDay
                     ? sdfTime.format(event.getEnd()) : sdf
                     .format(event.getEnd());

                json = json + '{ "eventId" : "' + event.getId()
                     + '", "eventTitle" : "' + event.getTitle()
                     + '", "eventStart" : "' + startEvent
                     + '", "eventEnd" : "' + endEvent
                     + '", "eventDateCreation" : "'
                     + sdfTime.format(event.getDateCreation())
                     + '", "eventDateUse" : "'
                     + sdfTime.format(event.getDateUse()) + '"}';
           }
      }
} catch (e) {
 }

 json = json + ' ] })';
```

```
    return json;
}
```

ENUMERATEEVENTS (EXEMPLE 2)

```
function enumerateEvents2() {
    var events = new java.util.ArrayList(0);

    var sdfTime = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy");

    var calendar = java.util.Calendar.getInstance();
    calendar.set(java.util.Calendar.HOUR_OF_DAY, 0);
    calendar.set(java.util.Calendar.MINUTE, 0);
    calendar.set(java.util.Calendar.SECOND, 0);
    calendar.set(java.util.Calendar.MILLISECOND, 0);

    var startDate = sdfTime.format(calendar.getTime());
    var dto = DTOFactory.newDTO("start", startDate);

    var json = '({ "eventList" : [ ';

    try {
          Calendar.enumerateEvents(dto, events);

          if (events != null && !events.isEmpty()) {
               for (var i = 0; i < events.size(); i++) {
                    var event = events.get(i);

                    if (i > 0) {
                         json = json + ', ';
                    }

                    var allDay = event.getAllDay() == 1 ? true : false;
                    var startEvent = allDay
```

```
                        ? sdfTime.format(event.getStart())
                        : sdf.format(event.getStart());
                    var endEvent = allDay
                        ? sdfTime.format(event.getEnd())
                        : sdf .format(event.getEnd());

                    json = json + '{ "eventId" : "' + event.getId()
                        + '", "eventTitle" : "' + event.getTitle()
                        + '", "eventStart" : "' + startEvent
                        + '", "eventEnd" : "' + endEvent
                        + '", "eventDateCreation" : "'
                        + sdfTime.format(event.getDateCreation())
                        + '", "eventDateUse" : "'
                        + sdfTime.format(event.getDateUse()) + '"}';
                }
            }
    } catch (e) {
     }

    json = json + ' ] })';

    return json;
}
```

ENUMERATEEVENTS (EXEMPLE 3)

```
function enumerateEvents3() {
    var events = new java.util.ArrayList(0);

    var sdfTime = new java.text.SimpleDateFormat("dd/MM/yyyy HH:mm");
    var sdf = new java.text.SimpleDateFormat("dd/MM/yyyy");

    var tagName = "TAG_GROUP_MDA";
    var tagId = Calendar.idOfTag(tagName);
    var dto = DTOFactory.newDTO("tagIds", tagId.toString());
```

```
var json = '({ "eventList" : [ ';


try {
      Calendar.enumerateEvents(dto, events);


      if (events != null && !events.isEmpty()) {
           for (var i = 0; i < events.size(); i++) {
                var event = events.get(i);


                if (i > 0) {
                     json = json + ', ';
                }


                var allDay = event.getAllDay() == 1 ? true : false;
                var startEvent = allDay
                     ? sdfTime.format(event.getStart())
                     : sdf.format(event.getStart());
                var endEvent = allDay
                     ? sdfTime.format(event.getEnd()) :
                     sdf .format(event.getEnd());


                json = json + '{ "eventId" : "' + event.getId()
                     + '", "eventTitle" : "' + event.getTitle()
                     + '", "eventStart" : "' + startEvent
                     + '", "eventEnd" : "' + endEvent
                     + '", "eventDateCreation" : "'
                      + sdfTime.format(event.getDateCreation())
                     + '", "eventDateUse" : "'
                     + sdfTime.format(event.getDateUse()) + '"}';
           }
      }
} catch (e) {
 }


json = json + ' ] })';


return json;
```

```
}
```