LightningScheduler Class

Apex Reference Guide / ConnectApi Namespace / LightningScheduler Class

LightningScheduler Class

Create and update service appointments.

Namespace

ConnectApi

LightningScheduler Methods

These methods are for `LightningScheduler`. All methods are static.

- createServiceAppointment(createServiceAppointmentInput)
  Create a service appointment.
- updateServiceAppointment(updateServiceAppointmentInput)
  Update a service appointment.

createServiceAppointment(createServiceAppointmentInput)

Create a service appointment.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.ServiceAppointmentOutput
createServiceAppointment(ConnectApi.CreateServiceAppointmentInput
createServiceAppointmentInput)
```

Parameters

createServiceAppointmentInput

Type: ConnectApi.CreateServiceAppointmentInput

Input parameters to create a service appointment.

Return Value

Type: ConnectApi.ServiceAppointmentOutput

Usage

Considerations for using engagement channel types with the `service-appointments` resource:

- Enable **Schedule Appointments Using Engagement Channels** in Salesforce Scheduler Settings in your Salesforce org.
- When you create or modify appointments, shifts must be defined in the scheduling policy. For more information on setting up shifts in the scheduling policy, see Define Shift Rules in Scheduling Policy.

- When you use engagement channel type and shifts to create a service appointment, Salesforce Scheduler considers the default value for the Appointment Type (if not specified). However, Salesforce Scheduler only considers the engagement channel type and Appointment Type is ignored.

Example

For an account (existing user):

```
ConnectApi.ExtendedFieldInput extendedFieldEmail = new ConnectApi.ExtendedFieldInput();
extendedFieldEmail.name = 'Email';
extendedFieldEmail.value = 'rachael.adams@salesforce.com';

ConnectApi.ExtendedFieldInput extendedFieldPhone = new ConnectApi.ExtendedFieldInput();
extendedFieldPhone.name = 'Phone';
extendedFieldPhone.value = '1234567890';

List<ConnectApi.ExtendedFieldInput> extendedFieldList = new List<ConnectApi.ExtendedFieldI
extendedFieldList.add(extendedFieldEmail);
extendedFieldList.add(extendedFieldPhone);

ConnectApi.ServiceAppointmentInput serviceAppInput = new ConnectApi.ServiceAppointmentInpu
serviceAppInput.extendedFields = extendedFieldList;
serviceAppInput.engagementChannelTypeId = '0eFRM00000000Bv2AI';
serviceAppInput.serviceTerritoryId = '0Hhxx0000004C92CAE';
serviceAppInput.workTypeId = '08qxx0000004C92AAE';
serviceAppInput.parentRecordId = '001xx000003GYR1AAO';
serviceAppInput.schedStartTime = DateTime.valueOf('2021-05-28 12:15:00');
serviceAppInput.schedEndTime = DateTime.valueOf('2021-05-28 12:45:00');
serviceAppInput.appointmentMode = 'Group';
serviceAppInput.attendeeLimit = 20;

ConnectApi.AssignedResourcesInput asResourceInput = new ConnectApi.AssignedResourcesInput(
asResourceInput.serviceResourceId = '0Hnxx0000004CAiCAM';
asResourceInput.isRequiredResource = true;
asResourceInput.isPrimaryResource = true;

List<ConnectApi.AssignedResourcesInput> asResourceInputList = new List<ConnectApi.Assigned
asResourceInputList.add(asResourceInput);

ConnectApi.CreateServiceAppointmentInput createInput = new ConnectApi.CreateServiceAppoint
createInput.serviceAppointment = serviceAppInput;
createInput.assignedResources = asResourceInputList;

try{
   ConnectApi.ServiceAppointmentOutput appointmentResult = ConnectApi.LightningScheduler.c
   String serviceAppointmentId = appointmentResult.result.serviceAppointmentId;
   List<String> assignedResourceIds = appointmentResult.result.assignedResourceIds;
}catch(ConnectApi.ConnectApiException ex){
   //Handle Exception
}
```

For a lead (authenticated guest user):

```
ConnectApi.LeadInput leadInput = new ConnectApi.LeadInput();
leadInput.firstName = 'Rachel';
leadInput.lastName = 'Adams';
leadInput.phone = '012-345-6789';
leadInput.email = 'rachel.adams@salesforce.com';
leadInput.company = 'Salesforce';

ConnectApi.ExtendedFieldInput extendedFieldEmail = new ConnectApi.ExtendedFieldInput();
extendedFieldEmail.name = 'Email';
extendedFieldEmail.value = 'rachael.adams@salesforce.com';
```

```
    extendedFieldList.add(extendedFieldEmail);
    extendedFieldList.add(extendedFieldPhone);

    ConnectApi.ServiceAppointmentInput serviceAppInput = new ConnectApi.ServiceAppointmentInpu
    serviceAppInput.extendedFields = extendedFieldList;
    serviceAppInput.engagementChannelTypeId = '0eFRM00000000Bv2AI';
    serviceAppInput.serviceTerritoryId = '0Hhxx0000004C92CAE';
    serviceAppInput.workTypeId = '08qxx0000004C92AAE';
    serviceAppInput.schedStartTime = DateTime.valueOf('2021-05-28 12:15:00');
    serviceAppInput.schedEndTime = DateTime.valueOf('2021-05-28 12:45:00');

    ConnectApi.AssignedResourcesInput asResourceInput = new ConnectApi.AssignedResourcesInput(
    asResourceInput.serviceResourceId = '0Hnxx0000004CAiCAM';
    asResourceInput.isRequiredResource = true;
    asResourceInput.isPrimaryResource = true;

    List<ConnectApi.AssignedResourcesInput> asResourceInputList = new List<ConnectApi.Assigned
    asResourceInputList.add(asResourceInput);

    ConnectApi.CreateServiceAppointmentInput createInput = new ConnectApi.CreateServiceAppoint
    createInput.serviceAppointment = serviceAppInput;
    createInput.assignedResources = asResourceInputList;
    createInput.lead = leadInput;

    try{
       ConnectApi.ServiceAppointmentOutput appointmentResult = ConnectApi.LightningScheduler.c
       String serviceAppointmentId = appointmentResult.result.serviceAppointmentId;
       List<String> assignedResourceIds = appointmentResult.result.assignedResourceIds;
    }catch(ConnectApi.ConnectApiException ex){
       //Handle Exception
    }
```

See Also

- Service Appointments

updateServiceAppointment(updateServiceAppointmentInput)

Update a service appointment.

API Version

53.0

Requires Chatter

No

Signature

```
public static ConnectApi.ServiceAppointmentOutput
updateServiceAppointment(ConnectApi.UpdateServiceAppointmentInput
updateServiceAppointmentInput)
```

Parameters

updateServiceAppointmentInput

Type: ConnectApi.UpdateServiceAppointmentInput

Input parameters to update a service appointment.

Return Value

Type: ConnectApi.ServiceAppointmentOutput

- When you create or modify appointments, shifts must be defined in the scheduling policy. For more information on setting up shifts in the scheduling policy, see Define Shift Rules in Scheduling Policy.

> **Note**
>
> Engagement channel types are not supported with operating hours rules in the scheduling policy.

- When you use engagement channel type and shifts to modify an appointment, Salesforce Scheduler considers the default value for the Appointment Type (if not specified). However, Salesforce Scheduler only considers the engagement channel type and Appointment Type is ignored.

Example

```
ConnectApi.ExtendedFieldInput extendedFieldEmail = new ConnectApi.ExtendedFieldInput();
extendedFieldEmail.name = 'Email';
extendedFieldEmail.value = 'rachel.adams@salesforce.com.example';

ConnectApi.ExtendedFieldInput extendedFieldPhone = new ConnectApi.ExtendedFieldInput();
extendedFieldPhone.name = 'Phone';
extendedFieldPhone.value = '0123456789';

ConnectApi.ExtendedFieldInput extendedFieldStatus = new ConnectApi.ExtendedFieldInput();
extendedFieldStatus.name = 'Status';
extendedFieldStatus.value = 'None';

List<ConnectApi.ExtendedFieldInput> extendedFieldList = new List<ConnectApi.ExtendedFieldI
extendedFieldList.add(extendedFieldEmail);
extendedFieldList.add(extendedFieldPhone);
extendedFieldList.add(extendedFieldStatus);

ConnectApi.ServiceAppointmentInput serviceAppInput = new ConnectApi.ServiceAppointmentInpu
serviceAppInput.extendedFields = extendedFieldList;
serviceAppInput.serviceTerritoryId = '0Hhxx0000004C92CAE';
serviceAppInput.workTypeId = '08qxx0000004C92AAE';
serviceAppInput.schedStartTime = DateTime.valueOf('2021-05-28 12:15:00');
serviceAppInput.schedEndTime = DateTime.valueOf('2021-05-28 12:45:00');

ConnectApi.AssignedResourcesInput asResourceInput = new ConnectApi.AssignedResourcesInput(
asResourceInput.serviceResourceId = '0Hnxx0000004CAiCAM';
asResourceInput.isRequiredResource = true;
asResourceInput.isPrimaryResource = true;

//Multi-resource
ConnectApi.AssignedResourcesInput asResourceInputReq = new ConnectApi.AssignedResourcesInp
asResourceInputReq.serviceResourceId = '0Hnxx0000004CAgCAM';
asResourceInputReq.isRequiredResource = true;
asResourceInputReq.isPrimaryResource = false;

List<ConnectApi.AssignedResourcesInput> asResourceInputList = new List<ConnectApi.Assigned
asResourceInputList.add(asResourceInput);
asResourceInputList.add(asResourceInputReq);

ConnectApi.UpdateServiceAppointmentInput updateInput = new ConnectApi.UpdateServiceAppoint
updateInput.serviceAppointment = serviceAppInput;
updateInput.assignedResources = asResourceInputList;
updateInput.serviceAppointmentId = '08pxx0000004CYqAAM';

try{
   ConnectApi.ServiceAppointmentOutput appointmentResult = ConnectApi.LightningScheduler.u
   String serviceAppointmentId = appointmentResult.result.serviceAppointmentId;
   List<String> assignedResourceIds = appointmentResult.result.assignedResourceIds;
}catch(ConnectApi.ConnectApiException ex){
```

- Service Appointments

DID THIS ARTICLE SOLVE YOUR ISSUE?

Let us know so we can improve!

Share your feedback

DEVELOPER CENTERS
Heroku
MuleSoft
Tableau
Commerce Cloud
Lightning Design System
Einstein
Quip

POPULAR RESOURCES
Documentation
Component Library
APIs
Trailhead
Sample Apps
Podcasts
AppExchange

COMMUNITY
Trailblazer Community
Events and Calendar
Partner Community
Blog
Salesforce Admins
Salesforce Architects

Privacy Information Terms of Service Legal Use of Cookies Trust Cookie Preferences

Your Privacy Choices Responsible Disclosure Contact