Migration Guide: Switching from Concierge V1 API to V2 API

This guide will help you transition from the Concierge V1 API to the V2 API, providing equivalent methods and examples.

Object Reference Change

- **V1:** `window.capacityConcierge`
- **V2:** `window.CapacityConciergeActionV2`

Open/Close Concierge

Open

V1:

```
window.capacityConcierge.open();

```

V2:

```
window.CapacityConciergeActionV2.setInterface({ isOpen: true });

```

Close

V1:

```
window.capacityConcierge.close();

```

V2:

```
window.CapacityConciergeActionV2.setInterface({ isOpen: false });

```

Maximize

V1:

```
window.capacityConcierge.maximize();

```

V2:

```
window.CapacityConciergeActionV2.setInterface({ isOpen: false, size:
window.CapacityConciergeActionV2.InterfaceSizes.MAX });

```

Show/Hide CallButton

Show

V1:

```
window.capacityConcierge.show_callbutton();
window.capacityConcierge.hide_callbutton();
```

```
3
```

V2:

```
window.CapacityConciergeActionV2.setConciergeButton({ isShowing: true });
window.CapacityConciergeActionV2.setConciergeButton({ isShowing: false });

```

Hide

V1:

```
window.capacityConcierge.hide_callbutton();

```

V2:

```
window.CapacityConciergeActionV2.setConciergeButton({ isShowing: false });

```

Load/Submit Inquiries

Load (deprecated)

V1:

```
window.capacityConcierge.load('Hello');
window.capacityConcierge.load('Hello', true);

```

V2:

There is no direct equivalent for just loading text into the input field without submitting. Instead, you can directly submit an inquiry.

Submit (deprecated)

V1:

```
window.capacityConcierge.submit();

```

V2:

There is no direct equivalent for just submitting text from the input field. Instead, you can directly submit an inquiry (refer to Concierge 2.0 API Documentation for advance usage)

```
window.CapacityConciergeActionV2.sendInquiry({ input: 'Hello', tag: 'CUSTOM_INQUIRY' });

```

Ask Inquiry

V1:

```
window.capacityConcierge.ask('Hello');

```

V2:

```
window.CapacityConciergeActionV2.sendInquiry({ input: 'Hello', tag: 'CUSTOM_INQUIRY' });

```

Clear Inquiry (deprecated)

V1:

```
window.capacityConcierge.clear();

```

V2:

There is no direct equivalent for clearing text from the input field.

Stop Ongoing Conversation & Clear History

V1:

```
window.capacityConcierge.clear_history();

```

V2:

```
window.CapacityConciergeActionV2.clearHistory()

```

Bind Event Hooks

V1:

```
window.capacityConcierge.bind_event_hooks({
  general_event: t => console.log(`General Event: type: "${t.event}"`),
  message_event: t => console.log(`Message Event: type: "${t.event}" value: "${t.value}"${t.state ? ' state: '
+ JSON.stringify(t.state) : ''}`),
  livechat_event: t => console.log(`Livechat Event: type: "${t.event}"`)
});

```

V2:

The new API uses the `eventHook` option within the `CapacityConciergeConfigV2` object, and no longer needs to differentiate the event into `general_event`, `message_event`, `livechat_event`

```
window.CapacityConciergeConfigV2 = {
  eventHook: (event) => {
    console.log(`Event received: type: "${event.event}"; value: "${event.value}"; ${event.states ? ' states: '
+ JSON.stringify(event.states) : ''}`);
  }
};

```

Set Session Variables

V1:

```
window.capacityConcierge.set_attachment({
  firstname: 'John',
  lastname: 'Smith'
});

```

V2:

The new API uses the `sessionVar` option within the `CapacityConciergeConfigV2` object where multiple custom configs/hooks combine.

```
window.CapacityConciergeConfigV2 = {
    ...,
    sessionVar: {
        firstname: 'John',
        lastname: 'Smith'
    }
}

```

Check Interface State

V1:

```
if (window.capacityConcierge?.is_open) {
  window.capacityConcierge.load('Hello'); // Do something...
}

```

V2:

```
(async () => {
  const isInterfaceOpen = await window.CapacityConciergeActionV2.getIsInterfaceOpenAsync();
  if (isInterfaceOpen) {
    window.CapacityConciergeActionV2.sendInquiry({ input: 'Hello', tag: 'CUSTOM_INQUIRY' }); // Do something...
  }
})();

```

Check If Concierge Has Ongoing Conversation

V1:

```
if (window.capacityConcierge?._isBusy) {
  // Do something...
}

```

V2:

```
(async () => {
  const isBusy = await window.CapacityConciergeActionV2.getIsSessionBusyAsync();
  if (isBusy) {
    // Do something...
  }
})();

```

Check If Concierge Has Ongoing Livechat

V1:

```
if (window.capacityConcierge?._isLivechatActive) {
  // Do something...
}

```

V2:

```
(async () => {
  const isLivechatActive = await window.CapacityConciergeActionV2.getIsLivechatActiveAsync();
  if (isLivechatActive) {
    // Do something...
  }
})();

```

Set Avoid Element (deprecated)

V1:

```
window.capacityConcierge.set_avoid_element();

```

V2:

There is no direct equivalent for setting target element for Concierge to dodge, consider using different `corner` (Console - Concierge Setting), custom css, or custom corner layout instead. Refer to Concierge 2.0 API Documentation or consult with us.

Check If Input Field Is Populated (deprecated)

V1:

```
if (window.capacityConcierge?.has_message) {
  window.capacityConcierge.submit(); // Do something...
}

```

V2:

There is no direct parallel in V2 to check if a message exists in the input field, but you can handle message submission directly.

Additional Features in V2

Set Page

```
window.CapacityConciergeActionV2.setPage({ page: window.CapacityConciergeActionV2.InterfacePages.MAIN });
window.CapacityConciergeActionV2.setPage({ page: window.CapacityConciergeActionV2.InterfacePages.CHAT });

```

Set Interface Size

```
window.CapacityConciergeActionV2.setInterface({ isOpen: true, size:
window.CapacityConciergeActionV2.InterfaceSizes.MAX });
window.CapacityConciergeActionV2.setInterface({ isOpen: true, size:
window.CapacityConciergeActionV2.InterfaceSizes.NORMAL });

```

Stop current conversation

```
window.CapacityConciergeActionV2.stopSession()

```

Run Integration

```
window.CapacityConciergeActionV2.runIntegration({ key:
window.CapacityConciergeActionV2.IntegrationKeys.SALESFORCE_EMBEDDED_SERVICE, params: { ... } });

```

```
```

Restart Concierge

```
window.CapacityConciergeActionV2.restart()

```

Destroy Concierge

```
const CONCIERGE_TOKEN = '...'

window.CapacityConciergeActionV2.destroy({ token: CONCIERGE_TOKEN })

```

Get Current Page From Concierge

```
(async () => {
    const currentInterfacePage = await window.CapacityConciergeActionV2.getCurrentInterfacePageAsync()

    console.log(`Concierge interface is showing ${currentInterfacePage === window.CapacityConciergeActionV2.InterfacePages.CHAT ? 'chat': 'main'} page`)
})()

```

Set Callback For Concierge

```
window.CapacityConciergeConfigV2 = {
    ...,
    callback: () => {
        console.log('Concierge window.CapacityConciergeConfigV2.callback method is called.')
    }
}

```

Set Options For Concierge

```
window.CapacityConciergeConfigV2 = {
    ...,
    window.CapacityConciergeConfigV2 = {
      ...,
      option: {
          // Concierge Option here (Refer to the V2 API documentation)
      }
  }
}

```

Set Custom CSS For Concierge Elements

```
window.CapacityConciergeConfigV2 = {
    ...,
    css: `
.ccg2_teaser__bubble {
  background-color: #27374D;
}
.ccg2_teaser__text {
  color: #fff;
}

```

```
.ccg2_teaser__chip {
  box-shadow: 0px 0px 0px 1px #27374D;
  background-color: #27374D;
  color: #fff;
}
`
}

```

Refer to the comprehensive V2 API documentation for any additional features or detailed usage.

With this guide, you should be able to migrate your application from Concierge V1 API to Concierge V2 API with ease. If you encounter any issues during migration, please consult the detailed documentation or reach out.