

Ship the SDK

The 18 steps to ship the SDK, from your first line of code to an identified event in Live. Tick them off as you go.

- | | | |
|--------------------------|--|-------------|
| ☐ | 1. Decide your path: browser autocapture plus track() for web pages, or a server SDK or the Track Data API for anything with no browser. | Lesson 13.1 |
| ☐ | 2. In the console go to Integrations, add a source for your platform (Configure JavaScript, Node JS, Python, PHP, iOS, Android or React Native), name it, and copy the Installation Snippet and Source ID. | Lesson 13.2 |
| ☐ | 3. Get an API key in prefix.secret form from Settings, API keys, and keep it out of source control. | Lesson 13.3 |
| ☐ | 4. Browser: add the queue stub and <code><script async src="https://cdn.intempt.com/v1/intempt.min.js?organization=...&project=...&source=...&key=..."></code> to the page <code><head></code> . Test on a real domain, not localhost. | Lesson 13.2 |
| ☐ | 5. Browser: call <code>intempt.identify({ userId })</code> then <code>intempt.track({ eventName, data })</code> with non-empty data. | Lesson 13.2 |
| ☐ | 6. Node: <code>npm install intempt-nodejs-sdk</code> , then <code>Intempt.init({ org, project, apiKey, sourceId })</code> once per process, with <code>sourceId</code> as a string. | Lesson 13.3 |
| ☐ | 7. Node: <code>await intempt.identify({ userId, traits })</code> and <code>await intempt.track('event', { userId, properties })</code> ; call <code>await intempt.close()</code> at shutdown. | Lesson 13.3 |
| ☐ | 8. Python: install the intempt package, then <code>Intempt(org=..., project=..., api_key=..., source_id=...)</code> with <code>source_id</code> as a string. | Lesson 13.4 |
| ☐ | 9. Python: <code>intempt.identify(user_id=..., traits=...)</code> then <code>intempt.track("event", user_id=..., properties=...)</code> ; call <code>intempt.close()</code> at shutdown. | Lesson 13.4 |
| ☐ | 10. PHP: composer require intempt/intempt-php (PHP 8.1+, ext-curl, ext-json), then <code>new Intempt(['org' => ..., 'project' => ..., 'apiKey' => ..., 'sourceId' => ...])</code> . | Lesson 13.5 |
| ☐ | 11. PHP: <code>\$intempt->identify(['userId' => ..., 'traits' => ...])</code> then <code>\$intempt->track('event', ['userId' => ..., 'properties' => ...])</code> . | Lesson 13.5 |

-
- | | | |
|------------------------------|---|-------------|
| ☐ 12. | iOS: add <code>https://github.com/intempt/intempt-swift.git</code> from 0.4.0 via Swift Package Manager, or <code>pod 'Intempt', '~> 0.4.0'</code> ; then try <code>IntemptInstance.initialize(apiKey:orgId:projectId:sourceId:)</code> at launch (iOS 15+). | Lesson 13.6 |
|------------------------------|---|-------------|
-
- | | | |
|------------------------------|---|-------------|
| ☐ 13. | iOS: <code>intempt.identify(userId:)</code> then <code>intempt.track(eventTitle:data:)</code> . | Lesson 13.6 |
|------------------------------|---|-------------|
-
- | | | |
|------------------------------|--|-------------|
| ☐ 14. | Android: add <code>implementation("com.intempt.sdk:intempt-android:<version>")</code> , declare <code>android.permission.INTERNET</code> , put credentials in <code>src/main/assets/intempt-config.json</code> , and call <code>Intempt.initialize(this)</code> in <code>Application.onCreate()</code> . | Lesson 13.7 |
|------------------------------|--|-------------|
-
- | | | |
|------------------------------|---|-------------|
| ☐ 15. | Android: <code>Intempt.identify("user-id")</code> then <code>Intempt.track("Event name")</code> . | Lesson 13.7 |
|------------------------------|---|-------------|
-
- | | | |
|------------------------------|---|-------------|
| ☐ 16. | React Native: <code>npm install @intempt-technologies/react-native</code> , run <code>cd ios</code> && <code>pod install</code> , then <code>const intempt = await init({ apiKey, orgId, projectId, sourceId })</code> before the first screen. | Lesson 13.8 |
|------------------------------|---|-------------|
-
- | | | |
|------------------------------|--|-------------|
| ☐ 17. | React Native: <code>await intempt.identify('user-id', { userAttributes })</code> then <code>await intempt.track('Event', { ... })</code> . | Lesson 13.8 |
|------------------------------|--|-------------|
-
- | | | |
|------------------------------|--|-------------|
| ☐ 18. | Open Events, then the Live tab, and watch your event arrive. Do not skip the wait. | Lesson 13.2 |
|------------------------------|--|-------------|
-

YOU ARE DONE WHEN

- Your custom event appears on Events, Live in the console.
- The event is attached to the identified user's profile, not an anonymous one.
- Browser only: tested on a real domain, since tracking is blocked on localhost.
- Server SDKs: flush or close ran so buffered events were sent before the process exited.

CORE STACK

- **IntemptJS (CDN script tag)** This is the browser SDK. One script tag, nothing to install
- **intempt-nodejs-sdk** This is the SDK for Node servers
- **intempt (PyPI)** This is the SDK for Python
- **intempt/intempt-php** This is the SDK for PHP
- **Intempt (CocoaPods)** This is the SDK for iPhone and iPad apps
- **com.intempt.sdk:intempt-android** This is the SDK for Android apps
- **@intempt-technologies/react-native** This is the SDK for React Native apps

OPTIONAL UPGRADES

- **Track Data API** This is how you send events with no SDK at all
- **Autocapture and automatic events** This is what every SDK tracks for you, before you write any code

- **Java SDK** This is the SDK for Java, if that is your stack

Every SDK is free. Your first 100K events a month are included on every plan. · intempt.com/docs/academy