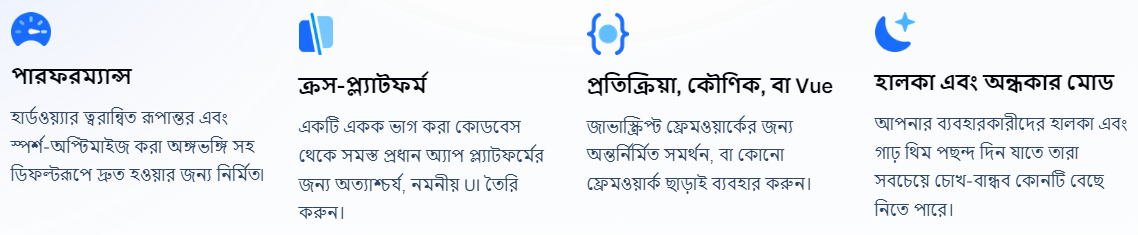
**Ionic Framework**

**আইওনিক ফ্রেমওয়ার্ক**

ওয়েবের জন্য মোবাইল এপ্লিকেশন তৈরি জন্য **Ionic Framework** ব্যবহার করা হয়।

ক্রস-প্ল্যাটফর্ম মোবাইল অ্যাপ: এটি মোবাইল অ্যাপ্লিকেশন বোঝায় যা একই কোড ব্যবহার করে একাধিক মোবাইল প্ল্যাটফর্মে (যেমন Android এবং iOS) চলানো যায়।

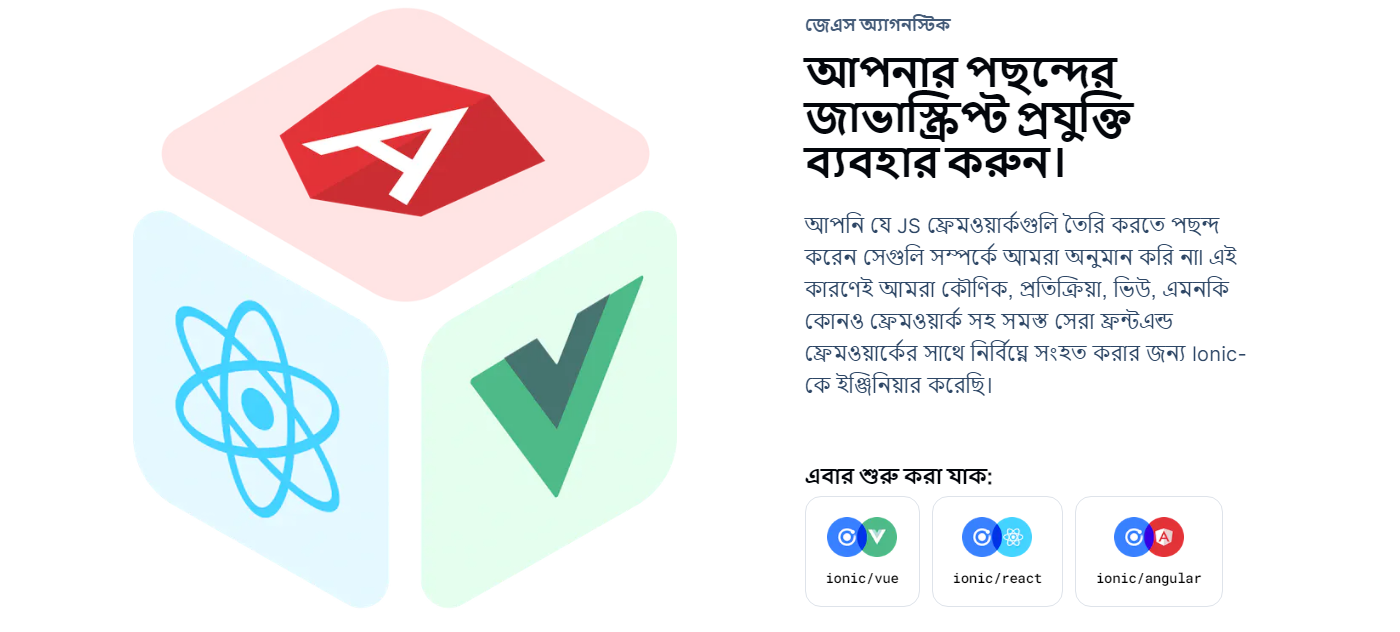
একটি একক কোড বেস থেকে আধুনিক, উচ্চ মানের [**ক্রস-প্ল্যাটফর্ম মোবাইল অ্যাপ**](ক্রস-প্ল্যাটফর্ম%20মোবাইল%20অ্যাপ.docx) তৈরি করার জন্য একটি ওপেন সোর্স মোবাইল UI টুলকিট হচ্ছে **আইওনিক ফ্রেমওয়ার্ক।**



[**হাইব্রিড এবং নেটিভ অ্যাপের মধ্যে পার্থক্য জানতে এখানে ক্লিক করুন Ctrl +Click।**](হাইব্রিড%20এবং%20নেটিভ%20অ্যাপের%20মধ্যে%20পার্থক্য.docx)

**ক্রস-প্ল্যাটফর্ম মোবাইল UI, ডিজাইন এপ**

আধুনিক UI দৃষ্টান্ত। মোবাইলের জন্য প্রস্তুত। মোবাইল এপ ডিজাইন করার জন্য আইওনিকের লাইব্রেরি আছে, সেখান থেকে অনেক ডিজাইন ব্যবহার করা যাবে। ডিজাইন লাইব্রেরি জন্য এখানে [**ক্লিক**](https://ionicframework.com/docs/components) করুন।



**Ionic অ্যাপস ডেভেলপ করার প্রসেস।**

**Ionic কমান্ড-লাইন ইন্টারফেস ( CLI ) হল Ionic অ্যাপস ডেভেলপ করার জন্য গো-টু টুল।**

## [ইনস্টলেশন](https://ionicframework.com/docs/v5/cli#installation)

আয়নিক সিএলআই এনপিএম এর সাথে বিশ্বব্যাপী ( গ্লোভালি ) ইনস্টল করা যেতে পারে:

npm install -g @ionic/cli

[Ionic CLI TypeScript](https://ionicframework.com/docs/reference/glossary#typescript) এবং [Node.js](https://ionicframework.com/docs/reference/glossary#node) দিয়ে নির্মিত । এটি নোড 10.3+ সমর্থন করে, তবে সর্বশেষ নোড এলটিএস সর্বদা সুপারিশ করা হয়। ওপেন সোর্স [গিটহাব রিপোজিটরিতে](https://github.com/ionic-team/ionic-cli) উন্নয়ন অনুসরণ করুন ।

## [সমস্যা সমাধান](https://ionicframework.com/docs/v5/cli#troubleshooting)

Ionic CLI এর সমস্যা সমাধানের জন্য, নিম্নলিখিতগুলি কার্যকর হতে পারে:

* Ionic CLI এর সর্বশেষ সংস্করণ ইনস্টল করা আছে তা নিশ্চিত করুন। রান করে ইন্সটল ভার্সনটি পান ionic --version।
* সর্বশেষ নোড LTS ইনস্টল করা আছে তা নিশ্চিত করুন। [নোড এবং এনপিএম](https://ionicframework.com/docs/intro/environment#node-npm) পরিবেশ সেটআপ দেখুন ।
* পতাকা --verboseডিবাগিং বার্তা প্রিন্ট করে, যা সমস্যাটিকে সংকুচিত করতে পারে।
* অনুপযুক্তভাবে কনফিগার করা প্রক্সি সেটিংসের কারণে সংযোগ সমস্যা হতে পারে৷ অনুরোধ প্রক্সি কনফিগার করতে [একটি প্রক্সি ব্যবহার করা](https://ionicframework.com/docs/cli/using-a-proxy) দেখুন ।
* গ্লোবাল Ionic CLI কনফিগারেশন ডিরেক্টরি ~/.ionicসমস্ত প্ল্যাটফর্মে রয়েছে। এটি নিরাপদে মুছে ফেলা যেতে পারে এবং Ionic CLI এটিকে পুনরুদ্ধার করবে, কিন্তু সমস্ত কনফিগারেশন (ব্যবহারকারীর সেশন সহ) হারিয়ে যাবে। [CLI এনভায়রনমেন্ট ভেরিয়েবল](https://ionicframework.com/docs/cli/configuration#environment-variables) দিয়ে এই ডিরেক্টরিটি কনফিগার করুন ।

**লাইভ রিলোড বা লাইভ ভিউ**

**মোবাইল ডিভাইসে এপ ভিউ করা।**

Ionic অ্যাপ তৈরি করার সময় উৎপাদনশীলতা বাড়াতে পারে এমন একটি বিকল্প হল **লাইভ রিলোড** (বা **লাইভ-রিলোড** )। অ্যাক্টিভ হলে, অ্যাপে পরিবর্তন শনাক্ত হলে লাইভ রিলোড ব্রাউজার বা [ওয়েব ভিউ রিলোড করবে।](https://ionicframework.com/docs/v5/core-concepts/webview)এটি হার্ডওয়্যার ডিভাইস ব্যবহার করে বিকাশের জন্য বিশেষভাবে কার্যকর।

## [শর্তাবলী](https://ionicframework.com/docs/v5/cli/livereload#terms)

লাইভ রিলোড একটি সংঘবদ্ধ শব্দ। এর সাথে ionic serve, লাইভ রিলোড মানে পরিবর্তন করা হলে ব্রাউজার পুনরায় লোড করা। ভার্চুয়াল এবং হার্ডওয়্যার ডিভাইসে একই অভিজ্ঞতা প্রদান করতে **ক্যাপাসিটর** এবং **কর্ডোভার** সাথে লাইভ রিলোডও ব্যবহার করা যেতে পারে, যা একটি নেটিভ বাইনারি পুনরায় স্থাপনের প্রয়োজনীয়তা দূর করে।

**আমরা এখানে কর্ডোভার ব্যবহার করে লাইভ ভিউ দেখবো।**

## [ব্যবহার](https://ionicframework.com/docs/v5/cli/livereload#usage)

যেহেতু লাইভ-রিলোডের জন্য ডিভাইসে ফাইলগুলি পড়ার পরিবর্তে আপনার কম্পিউটার দ্বারা হোস্ট করা একটি URL থেকে আপনার অ্যাপ লোড করতে ওয়েব ভিউ প্রয়োজন, তাই হার্ডওয়্যার ডিভাইসের জন্য লাইভ-রিলোড সেট আপ করা কঠিন হতে পারে।

নিয়মিত ডিভাইস স্থাপনের মতো, আপনার ডিভাইসটিকে আপনার কম্পিউটারে সংযুক্ত করতে আপনার একটি তারের প্রয়োজন হবে। পার্থক্য হল Ionic CLI আপনার কম্পিউটারে ডেভ সার্ভার থেকে আপনার অ্যাপ লোড করতে ওয়েব ভিউ কনফিগার করে।

### [কর্ডোভা](https://ionicframework.com/docs/v5/cli/livereload#cordova)

#### [অ্যান্ড্রয়েড](https://ionicframework.com/docs/v5/cli/livereload#android)

Android ডিভাইসের জন্য, Ionic CLI স্বয়ংক্রিয়ভাবে ডেভ সার্ভার পোর্ট ফরোয়ার্ড করবে। এর মানে হল আপনি একটি localhostঠিকানা ব্যবহার করতে পারেন এবং ওয়েব ভিউতে লোড করার সময় এটি আপনার কম্পিউটারে উল্লেখ করবে, ডিভাইসে নয়।

নিম্নলিখিত অল-ইন-ওয়ান কমান্ডটি একটি লাইভ-রিলোড সার্ভার চালু করবে localhostএবং কর্ডোভা ব্যবহার করে একটি অ্যান্ড্রয়েড ডিভাইসে অ্যাপটি স্থাপন করবে:

ionic cordova run android -l

কপি

#### [আইওএস](https://ionicframework.com/docs/v5/cli/livereload#ios)

iOS ডিভাইসের জন্য, পোর্ট ফরওয়ার্ডিং এখনও একটি বিকল্প নয়। এর মানে হল আপনাকে আপনার ডিভাইসটিকে আপনার কম্পিউটারের মতো একই Wi-Fi নেটওয়ার্কে সংযুক্ত করতে হবে এবং ডেভ সার্ভারের জন্য একটি বাহ্যিক ঠিকানা ব্যবহার করতে হবে৷

##### বিঃদ্রঃ

আপনি Ionic এর সাথে iOS পোর্ট ফরওয়ার্ডিং এর অগ্রগতির জন্য [এই সমস্যাটি ট্র্যাক করতে পারেন।](https://github.com/ionic-team/native-run/issues/20)

কিছু ক্ষেত্রে, Ionic CLI ঠিকানাটি জানে না যার সাথে ওয়েব ভিউ কনফিগার করতে হবে, তাই আপনাকে একটি নির্বাচন করতে বলা হতে পারে। আপনার Wi-Fi নেটওয়ার্কে আপনার কম্পিউটারের ঠিকানা নির্বাচন করতে ভুলবেন না।

**নিম্নলিখিত অল-ইন-ওয়ান কমান্ডটি সমস্ত ঠিকানায়** একটি লাইভ-রিলোড সার্ভার শুরু করবে এবং কর্ডোভা ব্যবহার করে একটি iOS ডিভাইসে অ্যাপটি স্থাপন করবে:

ionic cordova run ios -l --external

কপি

##### বিঃদ্রঃ

মনে রাখবেন, --externalবিকল্পটি সহ, আপনার Wi-Fi নেটওয়ার্কের অন্যরা আপনার অ্যাপ অ্যাক্সেস করতে সক্ষম হবে।

## [টিপস](https://ionicframework.com/docs/v5/cli/livereload#tips)

* কর্ডোভার সাথে, লক্ষ্য ডিভাইসগুলিকে সংকুচিত করতে , এবং বিকল্পগুলি --deviceব্যবহার করুন৷ সমস্ত লক্ষ্য তালিকার বিকল্পটি ব্যবহার করুন । [কমান্ড ডক্সে](https://ionicframework.com/docs/v5/cli/commands/cordova-run) ব্যবহার দেখুন ।--emulator--target--list
* ionic serveআপনি বা --livereload-urlএর বিকল্পটি ব্যবহার করে dev সার্ভার প্রক্রিয়া এবং স্থাপন প্রক্রিয়া আলাদা করতে ionic cordova runপারেন ionic capacitor run।
* অ্যান্ড্রয়েডের জন্য, অ্যাডবি সার্ভার চলাকালীন সবসময় পোর্ট ফরওয়ার্ড করার জন্য [অ্যাডবি](https://developer.android.com/studio/command-line/adb) কনফিগার করা সম্ভব (দেখুন adb reverse)। পোর্ট ফরওয়ার্ডিং সেট আপ সহ, একটি বহিরাগত ঠিকানা আর প্রয়োজন হবে না। এছাড়াও আপনি TCP এর উপর adb ব্রিজ সেটআপ করতে পারেন যাতে পরবর্তী স্থাপনার জন্য আর USB কেবলের প্রয়োজন হয় না।

# আয়নিক কর্ডোভা বিল্ড

Android এবং iOS প্ল্যাটফর্ম লক্ষ্যগুলির জন্য তৈরি করতে Cordova ব্যবহার করুন

ionic cordova build [options]

কপি

সরাসরি চালানোর মতো cordova build, ionic cordova buildএছাড়াও ওয়েব সম্পদ তৈরি করে ionic buildএবং Android এবং iOS প্ল্যাটফর্মের জন্য বন্ধুত্বপূর্ণ চেক প্রদান করে।

কর্ডোভা CLI-তে অতিরিক্ত বিকল্পগুলি পাস করতে, --Ionic CLI আর্গুমেন্টের পরে বিভাজক ব্যবহার করুন।

[অ্যান্ড্রয়েড বিল্ডগুলির](https://cordova.apache.org/docs/en/latest/guide/platforms/android/index.html#using-flags) জন্য প্ল্যাটফর্ম-নির্দিষ্ট আর্গুমেন্টের জন্য কর্ডোভা CLI-এর একটি বিভাজক প্রয়োজন , তাই Ionic CLI-এর জন্য একটি অতিরিক্ত বিভাজক প্রয়োজন, কিন্তু iOS [বিল্ডগুলির](https://cordova.apache.org/docs/en/latest/guide/platforms/ios/index.html#using-flags) জন্য এটির প্রয়োজন নেই ৷ বিভাজক সহ ব্যবহারের জন্য উদাহরণ কমান্ড দেখুন। পতাকা ব্যবহার এড়াতে, --buildConfigএকটি **build.json** ফাইল ব্যবহার করার কথা বিবেচনা করুন।

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/cordova-build#examples)

$ ionic cordova build android  
$ ionic cordova build android --buildConfig=build.json  
$ ionic cordova build android --prod --release -- -- --gradleArg=-PcdvBuildMultipleApks=true  
$ ionic cordova build android --prod --release -- -- --keystore=filename.keystore --alias=myalias  
$ ionic cordova build android --prod --release -- -- --minSdkVersion=21  
$ ionic cordova build android --prod --release -- -- --versionCode=55  
$ ionic cordova build android --prod --release --buildConfig=build.json  
$ ionic cordova build ios  
$ ionic cordova build ios --buildConfig=build.json  
$ ionic cordova build ios --prod --release  
$ ionic cordova build ios --prod --release -- --developmentTeam="ABCD" --codeSignIdentity="iPhone Developer" --packageType="app-store"  
$ ionic cordova build ios --prod --release --buildConfig=build.json

# আয়নিক কর্ডোভা অনুকরণ

একটি সিমুলেটর/এমুলেটরে একটি আয়নিক প্রকল্প অনুকরণ করুন

ionic cordova emulate [options]

কপি

আপনার অ্যাপ তৈরি করুন এবং এই কমান্ডটি ব্যবহার করে ডিভাইস এবং এমুলেটরগুলিতে এটি স্থাপন করুন। ঐচ্ছিকভাবে লাইভরিলোড কার্যকারিতার জন্য --livereloadডেভ সার্ভার ব্যবহার করার বিকল্পটি নির্দিষ্ট করুন।ionic serve

ionic buildএই কমান্ডটি প্রথমে ওয়েব সম্পদ তৈরি করতে ব্যবহার করবে (বা বিকল্পের ionic serveসাথে --livereload)। তারপর, cordova buildআপনার অ্যাপ কম্পাইল এবং প্রস্তুত করতে ব্যবহৃত হয়। অবশেষে, native-run [ইউটিলিটি](https://github.com/ionic-team/native-run) একটি ডিভাইসে আপনার অ্যাপ চালানোর জন্য ব্যবহার করা হয়। পরিবর্তে এই প্রক্রিয়ার জন্য কর্ডোভা ব্যবহার করতে, --no-native-runবিকল্পটি ব্যবহার করুন।

আপনার যদি একাধিক ডিভাইস এবং এমুলেটর থাকে, তাহলে আপনি বিকল্পের সাথে একটি নির্দিষ্টকে লক্ষ্য করতে পারেন --target। আপনি এর সাথে লক্ষ্যগুলি তালিকাভুক্ত করতে পারেন --list।

[Android এবং iOS-এর জন্য, আপনি এই ডক্সগুলি](https://ionicframework.com/docs/developer-resources/developer-tips) ব্যবহার করে ব্রাউজার ডেভেলপমেন্ট টুলের সাহায্যে আপনার ডিভাইসে রিমোট ডিবাগিং সেটআপ করতে পারেন ।

হার্ডওয়্যার ডিভাইসের সাথে ব্যবহার করার সময় --livereload, মনে রাখবেন যে লাইভরিলোড ডিভাইস এবং কম্পিউটারের মধ্যে একটি সক্রিয় সংযোগ প্রয়োজন। কিছু পরিস্থিতিতে, বিকল্পটি ব্যবহার করে আপনাকে একটি বহিরাগত ঠিকানায় ডেভ সার্ভার হোস্ট করতে হতে পারে --external। আরো তথ্যের জন্য এই [ডক্স](https://ionicframework.com/docs/cli/livereload) দেখুন .

ঠিক যেমন এর সাথে , আপনি বিভাজক ionic cordova buildব্যবহার করে কর্ডোভা CLI-তে অতিরিক্ত বিকল্পগুলি প্রেরণ করতে পারেন ৷ --ডেভ সার্ভারে অতিরিক্ত বিকল্পগুলি পাস করতে, ionic serveআলাদাভাবে ব্যবহার করার কথা বিবেচনা করুন এবং --livereload-urlবিকল্পটি ব্যবহার করুন।

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/cordova-emulate#examples)

$ ionic cordova emulate android  
$ ionic cordova emulate android --buildConfig=build.json  
$ ionic cordova emulate android --prod --release -- -- --gradleArg=-PcdvBuildMultipleApks=true  
$ ionic cordova emulate android --prod --release -- -- --keystore=filename.keystore --alias=myalias  
$ ionic cordova emulate android --prod --release -- -- --minSdkVersion=21  
$ ionic cordova emulate android --prod --release -- -- --versionCode=55  
$ ionic cordova emulate android --prod --release --buildConfig=build.json  
$ ionic cordova emulate android -l  
$ ionic cordova emulate ios  
$ ionic cordova emulate ios --buildConfig=build.json  
$ ionic cordova emulate ios --livereload --external  
$ ionic cordova emulate ios --livereload-url=http://localhost:8100  
$ ionic cordova emulate ios --prod --release  
$ ionic cordova emulate ios --prod --release -- --developmentTeam="ABCD" --codeSignIdentity="iPhone Developer" --packageType="app-store"  
$ ionic cordova emulate ios --prod --release --buildConfig=build.json

# আয়নিক কর্ডোভা প্ল্যাটফর্ম

কর্ডোভা প্ল্যাটফর্ম লক্ষ্যগুলি পরিচালনা করুন

ionic cordova platform [options]

কপি

সরাসরি চালানোর মতো cordova platform, কিন্তু ডিফল্ট আয়নিক আইকন এবং স্প্ল্যাশ স্ক্রিন সংস্থান (সময়ে add) যোগ করে এবং বন্ধুত্বপূর্ণ চেক প্রদান করে।

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/cordova-platform#examples)

$ ionic cordova platform  
$ ionic cordova platform add ios  
$ ionic cordova platform add android  
$ ionic cordova platform rm ios

# আয়নিক কর্ডোভা প্লাগইন

কর্ডোভা প্লাগইনগুলি পরিচালনা করুন৷

ionic cordova plugin [options]

কপি

সরাসরি দৌড়ানোর মত cordova plugin, কিন্তু বন্ধুত্বপূর্ণ চেক প্রদান করে।

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/cordova-plugin#examples)

$ ionic cordova plugin  
$ ionic cordova plugin add cordova-plugin-inappbrowser@latest  
$ ionic cordova plugin add phonegap-plugin-push --variable SENDER\_ID=XXXXX  
$ ionic cordova plugin rm cordova-plugin-camera

# আয়নিক কর্ডোভা প্রয়োজনীয়তা

প্ল্যাটফর্মগুলির জন্য সমস্ত প্রয়োজনীয়তা পরীক্ষা করে এবং মুদ্রণ করে

ionic cordova requirements [options]

কপি

সরাসরি দৌড়ানোর মত cordova requirements, কিন্তু বন্ধুত্বপূর্ণ চেক প্রদান করে।

## [ইনপুট](https://ionicframework.com/docs/v5/cli/commands/cordova-requirements#inputs)

| প্ল্যাটফর্ম | |
| --- | --- |
| **বর্ণনা** | যে প্ল্যাটফর্মের জন্য আপনি প্রয়োজনীয়তা সংগ্রহ করতে চান ( android, ios) |

# আয়নিক কর্ডোভা সম্পদ

স্বয়ংক্রিয়ভাবে আইকন এবং স্প্ল্যাশ স্ক্রিন সংস্থান তৈরি করুন

ionic cordova resources [options]

কপি

এই কমান্ডের সাহায্যে আপনার কর্ডোভা প্ল্যাটফর্মগুলির জন্য PNG উত্স চিত্রগুলি থেকে পুরোপুরি আকারের আইকন এবং স্প্ল্যাশ স্ক্রিন তৈরি করুন৷

আইকনগুলির জন্য উত্স চিত্রটি আদর্শভাবে কমপক্ষে **1024×1024px** হওয়া উচিত এবং এটি **রিসোর্স/icon.png** এ অবস্থিত । স্প্ল্যাশ স্ক্রিনের জন্য উত্স চিত্রটি আদর্শভাবে কমপক্ষে **2732×2732px** হওয়া উচিত এবং এটি **রিসোর্স/splash.png-** এ অবস্থিত । যদি আপনি ব্যবহার করেন , তাহলে **রিসোর্স/**ionic start ডিরেক্টরিতে ইতিমধ্যেই ডিফল্ট আয়নিক সংস্থান থাকা উচিত , যা আপনি ওভাররাইট করতে পারেন।

আপনি প্ল্যাটফর্ম-নির্দিষ্ট আইকন এবং স্প্ল্যাশ স্ক্রিনগুলিকে সংশ্লিষ্ট **সংস্থান/ < প্ল্যাটফর্ম > /** ডিরেক্টরিতে স্থাপন করে তৈরি করতে পারেন। উদাহরণস্বরূপ, অ্যান্ড্রয়েডের জন্য একটি আইকন তৈরি করতে, আপনার ইমেজটি **রিসোর্স/android/icon.png** এ রাখুন ।

সেরা ফলাফলের জন্য, স্প্ল্যাশ স্ক্রিনের আর্টওয়ার্কটি চিত্রের কেন্দ্রে একটি বর্গাকার ( **1200×1200px ) মধ্যে মোটামুটি ফিট হওয়া উচিত।**আপনি আপনার স্প্ল্যাশ স্ক্রিনের জন্য একটি টেমপ্লেট হিসাবে [**https://code.ionicframework.com/resources/splash.psd ব্যবহার করতে পারেন।**](https://code.ionicframework.com/resources/splash.psd)

ionic cordova resources**স্বয়ংক্রিয়ভাবে আপনার config.xml** আপডেট করবে জেনারেট করা ছবিগুলির পরিবর্তনগুলিকে প্রতিফলিত করতে, যা কর্ডোভা তারপর কনফিগার করে৷

এই কমান্ড স্থানীয়ভাবে সংস্থান তৈরি করতে cordova-res [ইউটিলিটি ব্যবহার করে।](https://github.com/ionic-team/cordova-res)

কর্ডোভা রেফারেন্স ডকুমেন্টেশন:

* আইকন: [**https://cordova.apache.org/docs/en/latest/config\_ref/images.html**](https://cordova.apache.org/docs/en/latest/config_ref/images.html)
* স্প্ল্যাশ স্ক্রিন: [**https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-splashscreen/**](https://cordova.apache.org/docs/en/latest/reference/cordova-plugin-splashscreen/)

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/cordova-resources#examples)

$ ionic cordova resources  
$ ionic cordova resources ios  
$ ionic cordova resources android

# আয়নিক কর্ডোভা রান

একটি সংযুক্ত ডিভাইসে একটি আয়নিক প্রকল্প চালান

ionic cordova run [options]

কপি

আপনার অ্যাপ তৈরি করুন এবং এই কমান্ডটি ব্যবহার করে ডিভাইস এবং এমুলেটরগুলিতে এটি স্থাপন করুন। ঐচ্ছিকভাবে লাইভরিলোড কার্যকারিতার জন্য --livereloadডেভ সার্ভার ব্যবহার করার বিকল্পটি নির্দিষ্ট করুন।ionic serve

ionic buildএই কমান্ডটি প্রথমে ওয়েব সম্পদ তৈরি করতে ব্যবহার করবে (বা বিকল্পের ionic serveসাথে --livereload)। তারপর, cordova buildআপনার অ্যাপ কম্পাইল এবং প্রস্তুত করতে ব্যবহৃত হয়। অবশেষে, native-run [ইউটিলিটি](https://github.com/ionic-team/native-run) একটি ডিভাইসে আপনার অ্যাপ চালানোর জন্য ব্যবহার করা হয়। পরিবর্তে এই প্রক্রিয়ার জন্য কর্ডোভা ব্যবহার করতে, --no-native-runবিকল্পটি ব্যবহার করুন।

আপনার যদি একাধিক ডিভাইস এবং এমুলেটর থাকে, তাহলে আপনি বিকল্পের সাথে একটি নির্দিষ্টকে লক্ষ্য করতে পারেন --target। আপনি এর সাথে লক্ষ্যগুলি তালিকাভুক্ত করতে পারেন --list।

[Android এবং iOS-এর জন্য, আপনি এই ডক্সগুলি](https://ionicframework.com/docs/developer-resources/developer-tips) ব্যবহার করে ব্রাউজার ডেভেলপমেন্ট টুলের সাহায্যে আপনার ডিভাইসে রিমোট ডিবাগিং সেটআপ করতে পারেন ।

হার্ডওয়্যার ডিভাইসের সাথে ব্যবহার করার সময় --livereload, মনে রাখবেন যে লাইভরিলোড ডিভাইস এবং কম্পিউটারের মধ্যে একটি সক্রিয় সংযোগ প্রয়োজন। কিছু পরিস্থিতিতে, বিকল্পটি ব্যবহার করে আপনাকে একটি বহিরাগত ঠিকানায় ডেভ সার্ভার হোস্ট করতে হতে পারে --external। আরো তথ্যের জন্য এই [ডক্স](https://ionicframework.com/docs/cli/livereload) দেখুন .

ঠিক যেমন এর সাথে , আপনি বিভাজক ionic cordova buildব্যবহার করে কর্ডোভা CLI-তে অতিরিক্ত বিকল্পগুলি প্রেরণ করতে পারেন ৷ --ডেভ সার্ভারে অতিরিক্ত বিকল্পগুলি পাস করতে, ionic serveআলাদাভাবে ব্যবহার করার কথা বিবেচনা করুন এবং --livereload-urlবিকল্পটি ব্যবহার করুন।

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/cordova-run#examples)

$ ionic cordova run android  
$ ionic cordova run android --buildConfig=build.json  
$ ionic cordova run android --prod --release -- -- --gradleArg=-PcdvBuildMultipleApks=true  
$ ionic cordova run android --prod --release -- -- --keystore=filename.keystore --alias=myalias  
$ ionic cordova run android --prod --release -- -- --minSdkVersion=21  
$ ionic cordova run android --prod --release -- -- --versionCode=55  
$ ionic cordova run android --prod --release --buildConfig=build.json  
$ ionic cordova run android -l  
$ ionic cordova run ios  
$ ionic cordova run ios --buildConfig=build.json  
$ ionic cordova run ios --livereload --external  
$ ionic cordova run ios --livereload-url=http://localhost:8100  
$ ionic cordova run ios --prod --release  
$ ionic cordova run ios --prod --release -- --developmentTeam="ABCD" --codeSignIdentity="iPhone Developer" --packageType="app-store"  
$ ionic cordova run ios --prod --release --buildConfig=build.json

# আয়নিক স্থাপন যোগ

প্রজেক্টে Appflow Deploy যোগ করে

ionic deploy add [options]

কপি

cordova-plugin-ionicএই কমান্ডটি ক্যাপাসিটর এবং কর্ডোভা উভয় প্রকল্পের জন্য Appflow Deploy প্লাগইন ( ) যোগ করে ।

ক্যাপাসিটর প্রকল্পগুলির জন্য এটি প্লাগইন ইনস্টল করার জন্য প্রয়োজনীয় সমস্ত পদক্ষেপগুলি চালায়, নেটিভ প্রকল্পগুলির সাথে সিঙ্ক করতে এবং সঠিক iOS এবং Android কনফিগারেশন ফাইলগুলিতে কনফিগারেশন যোগ করে৷

কর্ডোভা প্রকল্পগুলির জন্য এটি জমা দেওয়া পরামিতিগুলির সাথে সঠিক কর্ডোভা CLI কমান্ড চালানোর যত্ন নেয়।

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/deploy-add#examples)

$ ionic deploy add  
$ ionic deploy add --app-id=abcd1234 --channel-name="Master" --update-method=background  
$ ionic deploy add --max-store=2 --min-background-duration=30  
$ ionic deploy add --app-id=abcd1234 --channel-name="Master" --update-method=background -

# আয়নিক ডাক্তার পরীক্ষা

আপনার আয়নিক প্রকল্পের স্বাস্থ্য পরীক্ষা করুন

ionic doctor check [options]

কপি

এই কমান্ডটি সাধারণ সমস্যাগুলি সনাক্ত করে এবং মুদ্রণ করে এবং সেগুলি সমাধান করার জন্য পদক্ষেপগুলির পরামর্শ দেয়৷

কিছু সমস্যা স্বয়ংক্রিয়ভাবে ঠিক করা যেতে পারে। দেখুন ionic doctor treat --help\_

ঐচ্ছিকভাবে idএকটি একক সমস্যা চেক করার জন্য যুক্তি সরবরাহ করুন। ionic doctor listসমস্ত পরিচিত সমস্যা তালিকা ব্যবহার করুন .

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/doctor-check#examples)

$ ionic doctor check  
$ ionic doctor check git-not-used

কপি

## [ইনপুট](https://ionicframework.com/docs/v5/cli/commands/doctor-check#inputs)

| আইডি | |
| --- | --- |
| **বর্ণনা** | সমস্যা শনাক্তকারী |

# আয়নিক ডাক্তারের তালিকা

সমস্ত সমস্যা এবং তাদের শনাক্তকারীদের তালিকা করুন

ionic doctor list [options]

কপি

সমস্যার বিভিন্ন ট্যাগ থাকতে পারে:

* **চিকিত্সাযোগ্য** : ionic doctor treatসমস্যাটি সমাধান করার চেষ্টা করতে পারে
* **উপেক্ষা করা হয়েছে** : ionic doctor checkবা এর মধ্যে সনাক্ত না করার জন্য কনফিগার করা হয়েছেionic doctor treat
* **explicit-detection** : সমস্যাটি শুধুমাত্র এর সাথে স্পষ্টভাবে সনাক্ত করা হয়ionic doctor check <issue-id>

এই কমান্ডের সাথে তালিকাভুক্ত একটি আইডি মেলে ionic config set -g doctor.issues.<issue-id>.ignored true/falseযেখানে আপনি ব্যবহার করে একটি সমস্যা উপেক্ষা করা হয় বা না হয় তা ফ্লিপ করতে পারেন ।<issue-id>

# ionic ডাক্তার চিকিত্সা

আপনার আয়নিক প্রকল্পে সমস্যা সমাধানের চেষ্টা করুন

ionic doctor treat [options]

কপি

এই কমান্ডটি সনাক্ত করে এবং সাধারণ সমস্যাগুলি সমাধান করার চেষ্টা করে। একটি ফিক্স করার চেষ্টা করার আগে, পদক্ষেপগুলি মুদ্রিত হয় এবং একটি নিশ্চিতকরণ প্রম্পট প্রদর্শিত হয়।

ঐচ্ছিকভাবে idএকটি একক সমস্যা সমাধান করার চেষ্টা করার জন্য যুক্তি সরবরাহ করুন। ionic doctor listসমস্ত পরিচিত সমস্যা তালিকা ব্যবহার করুন .

## [উদাহরণ](https://ionicframework.com/docs/v5/cli/commands/doctor-treat#examples)

$ ionic doctor treat  
$ ionic doctor treat git-not-used

কপি

## [ইনপুট](https://ionicframework.com/docs/v5/cli/commands/doctor-treat#inputs)

| আইডি | |
| --- | --- |
| **বর্ণনা** | সমস্যা শনাক্তকারী |