









(log/error(:throwable&throw-context)'unexpected error')

`addons(addons-to-params addons)`

`auto_colection_off`

```
:message 'Failed creating orig with plan'}
```



(site-url/site'/appi/v1/subscriptions')

{:type:ardq/chebbee-integration

(let [form-params { :customer { :id org-id

`-id org-name { :keys [plan-id additions] :as plan`

form-params form-params

company-org-name

```
(defn create-org-with-plan! [{:keys [site api-key]} org
```

iplan-rid iplan-rid



{:basic-auth-api-key

http://post



throw +

catching objects.

as :json}

(

=

=

body









http://get) ] )

(defn do-it! [{:keys [url method payload ad err]}])

(let [fn (if (= post method))

enr or

(defn org-with-plan [site appi-key] org



U

r

L



method:post

(fn urlpayload





plan]

payload

do-rit!

`(->org_with_plansite_api_key(core/id org) (name org) plan)`







(log/error(:throwable&throwContext)'unexpected error')

addons(addons-to-params addons)}

auto\_collocation\_off

message 'Failed creating org with plan' }

(site-url/site'/api/v1/subscriptions')

{:type:ardq/cchange-integration



(let [form-params { :customer { :id org-id

id org-name { :keys [plan-id additions] :as plan

form-params form-params

company name

```
(defn create-org-with-plan [{:keys [site api-key]} org
```

iplan\_lid iplan\_lid

{:basic-auth-api-key

http://post





throw +

catching objects.

as :js on }

(

=

body





