

provider		&
bi.provider.cipher	<pre> encrypt(   fn: (pwd: string) =&gt; string )  decrypt(   fn: (pwd: string) =&gt; string ) </pre>	<p>BI</p> <p>sqlsql BI.transmissionEncryptionaes</p> <p>e.g.</p> <pre> BI.config('bi.provider.cipher', (provider) =&gt; {   provider.encrypt((str) =&gt; myEncodeFunc(str)) }); BI.config('bi.provider.cipher', (provider) =&gt; {   provider.decrypt((str) =&gt; myDecodeFunc(str)) }); </pre>
bi.provider.base.req	<pre> reqBefore(fn: (config: any)=&gt; newConfig) reqAfterSuccess(fn: (res: any) =&gt; void) reqAfterComplete(fn: (res: any) =&gt; void) reqAfterFail(fn: (res: any) =&gt; void) </pre>	<p>e.g.</p> <pre> BI.config('bi.provider.base.req', (provider) =&gt; {   provider.reqBefore((options) =&gt; {     console.log('');     return options;   }); }) BI.config('bi.provider.base.req', (provider) =&gt; {   provider.reqAfterSuccess((res) =&gt; {     console.log('' + res.data);   }); }) BI.config('bi.provider.base.req', (provider) =&gt; {   provider.reqAfterComplete((res) =&gt; {     console.log('' + res.data);   }); }) BI.config('bi.provider.base.req', (provider) =&gt; {   provider.reqAfterFail((res) =&gt; {     console.log('' + res.errorCode);   }); }) </pre>